



THE UNIVERSITY *of* EDINBURGH

Edinburgh Research Explorer

A Preconditioner for a Primal-Dual Newton Conjugate Gradients Method for Compressed Sensing Problems

Citation for published version:

Dassios, I, Fountoulakis, K & Gondzio, J 2015, 'A Preconditioner for a Primal-Dual Newton Conjugate Gradients Method for Compressed Sensing Problems' SIAM Journal on Scientific Computing, vol. 37, no. 6, pp. A2783-A2812. DOI: 10.1137/141002062

Digital Object Identifier (DOI):

[10.1137/141002062](https://doi.org/10.1137/141002062)

Link:

[Link to publication record in Edinburgh Research Explorer](#)

Document Version:

Peer reviewed version

Published In:

SIAM Journal on Scientific Computing

General rights

Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy

The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact openaccess@ed.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.



A PRECONDITIONER FOR A PRIMAL-DUAL NEWTON CONJUGATE GRADIENTS METHOD FOR COMPRESSED SENSING PROBLEMS

December 30, 2014, revised in July 2, 2015

IOANNIS DASSIOS*, KIMON FOUNTOLAKIS†, AND JACEK GONDZIO‡

Abstract. In this paper we are concerned with the solution of Compressed Sensing (CS) problems where the signals to be recovered are sparse in coherent and redundant dictionaries. We extend the primal-dual Newton Conjugate Gradients (pdNCG) method in [11] for CS problems. We provide an inexpensive and provably effective preconditioning technique for linear systems using pdNCG. Numerical results are presented on CS problems which demonstrate the performance of pdNCG with the proposed preconditioner compared to state-of-the-art existing solvers.

Key words. compressed sensing, ℓ_1 -analysis, total-variation, second-order methods, Newton conjugate gradients

1. Introduction. CS is concerned with recovering a signal $\tilde{x} \in \mathbb{R}^n$ by observing a linear combination of the signal

$$\tilde{b} = A\tilde{x},$$

where $A \in \mathbb{R}^{m \times n}$ is an under-determined linear operator with $m < n$ and $\tilde{b} \in \mathbb{R}^m$ are the observed measurements. Although this system has infinitely many solutions, reconstruction of \tilde{x} is possible due to its assumed properties. In particular, \tilde{x} is assumed to have a sparse image through a coherent and redundant dictionary $W \in E^{n \times l}$, where $E = \mathbb{R}$ or \mathbb{C} and $n \leq l$. More precisely, $W^*\tilde{x}$, is sparse, i.e. it has only few non-zero components, where the star superscript denotes the conjugate transpose. If $W^*\tilde{x}$ is sparse, under certain conditions on matrices A and W (discussed in Subsection 1.2) the optimal solution of the linear problem

$$\text{minimize } \|W^*x\|_1, \quad \text{subject to: } Ax = \tilde{b}$$

is \tilde{x} , where $\|\cdot\|_1$ is the ℓ_1 -norm.

Frequently measurements \tilde{b} might be contaminated with noise, i.e. one measures $b = \tilde{b} + e$ instead, where e is a vector of noise, usually modelled as Gaussian with zero-mean and bounded Euclidean norm. In addition, in realistic applications, $W^*\tilde{x}$ might not be exactly sparse, but its mass might be concentrated only on few of its components, while the rest are rapidly decaying. In this case, (again under certain conditions on matrices A and W) the optimal solution of the following problem

$$(1.1) \quad \text{minimize } f_c(x) := c\|W^*x\|_1 + \frac{1}{2}\|Ax - b\|_2^2,$$

is proved to be a good approximation to \tilde{x} . In (1.1), c is an a-priori chosen positive scalar and $\|\cdot\|_2$ is the Euclidean norm.

*I. Dassios is with the School of Mathematics and Maxwell Institute, The University of Edinburgh, Peter Guthrie Tait Road, Edinburgh EH9 3FD, United Kingdom e-mail: idassios@ed.ac.uk. I. Dassios is supported by EPSRC Grant EP/I017127/1

†K. Fountoulakis is with the School of Mathematics and Maxwell Institute, The University of Edinburgh, Peter Guthrie Tait Road, Edinburgh EH9 3FD, United Kingdom e-mail: K.Fountoulakis@sms.ed.ac.uk.

‡J. Gondzio is with the School of Mathematics and Maxwell Institute, The University of Edinburgh, Peter Guthrie Tait Road, Edinburgh EH9 3FD, United Kingdom e-mail: J.Gondzio@ed.ac.uk.

1.1. Brief description of CS applications. An example of W being redundant and coherent with orthonormal rows is the curvelet frame where an image is assumed to have an approximately sparse representation [8]. Moreover, for radar and sonar systems it is frequent that Gabor frames are used in order to reconstruct pulse trains from CS measurements [21]. For more applications a small survey is given in [9]. Isotropic Total-Variation (iTV) is another application of CS, which exploits the fact that digital images frequently have slowly varying pixels, except along edges. This property implies that digital images with respect to the discrete nabla operator, i.e. local differences of pixels, are approximately sparse. For iTV applications, matrix $W \in \mathbb{C}^{n \times n}$ is square, complex and rank-deficient with $\text{rank}(W) = n - 1$. An alternative to iTV is ℓ_1 -analysis, where matrix W is a Haar wavelet transform. However, a more pleasant to the eye reconstruction result is obtained by solving the iTV problem compared to the ℓ_1 -analysis problem, see [23].

1.2. Conditions and properties of CS matrices. There has been an extensive amount of literature studying conditions and properties of matrices A and W which guarantee recoverability of a good approximation of \tilde{x} by solving problem (1.1). For a thorough analysis we refer the reader to [7, 9, 23]. The previously cited papers use a version of the well-known *Restricted Isometry Property* (RIP) [9], which is repeated below.

DEFINITION 1.1. *The restricted isometry constant of a matrix $A \in \mathbb{R}^{m \times n}$ adapted to $W \in E^{n \times l}$ is defined as the smallest δ_q such that*

$$(1 - \delta_q)\|Wz\|_2^2 \leq \|AWz\|_2^2 \leq (1 + \delta_q)\|Wz\|_2^2$$

for all at most q -sparse $z \in E^l$, where $E = \mathbb{R}$ or \mathbb{C} .

For the rest of the paper we will refer to Definition 1.1 as W-RIP. It is proved in Theorem 1.4 in [9] that if $W \in E^{n \times l}$ has orthonormal rows with $n \leq l$ and if A , W satisfy the W-RIP with $\delta_{2q} \leq 8.0e-2$, then the solution x_c obtained by solving problem (1.1) satisfies

$$(1.2) \quad \|x_c - \tilde{x}\|_2 = C_0\|e\|_2 + C_1 \frac{\|W^*x_c - (W^*\tilde{x})_q\|_1}{\sqrt{q}},$$

where $(W^*\tilde{x})_q$ is the best q -sparse approximation of $W^*\tilde{x}$, C_0 and C_1 are small constants and only depend on δ_{2q} . It is clear that $W^*\tilde{x}$ must have $l - q$ rapidly decaying components, in order for $\|x_c - \tilde{x}\|_2$ to be small and the reconstruction to be successful.

iTV is a special case of ℓ_1 -analysis where matrix W does not have orthonormal rows, hence, result (1.2) does not hold. For iTV there are no conditions on δ_{2q} such that a good reconstruction is assured. However, there exist results which directly impose restrictions on the number of measurements m , see Theorems 2, 5 and 6 in [23]. Briefly, in these theorems it is mentioned that if $m \geq q \log(n)$ linear measurements are acquired for which matrices A and W satisfy the W-RIP for some $\delta_q < 1$, then, similar reconstruction guarantees as in (1.2) are obtained for iTV. Based on the previously mentioned results regarding reconstruction guarantees it is natural to assume that for iTV a similar condition applies, i.e. $\delta_{2q} < 1/2$. Hence, we make the following assumption.

ASSUMPTION 1.2. *The number of nonzero components of W^*x_c , denoted by q , and the dimensions l , m , n are such that matrices A and W satisfy W-RIP for some $\delta_{2q} < 1/2$. This assumption will be used in the spectral analysis of our preconditioner in Section 5.*

Another property of matrix A is the near orthogonality of its rows. Indeed many applications in CS use matrices A that satisfy

$$(1.3) \quad \|AA^\top - I_m\|_2 \leq \delta,$$

with a small constant $\delta \geq 0$. Finally, through the paper we will make use of the following assumption

$$(1.4) \quad \text{Ker}(W^*) \cap \text{Ker}(A) = \{0\},$$

This is commonly used assumption in the literature, see for example [26].

1.3. Contribution. In [11], Chan, Golub and Mulet, proposed a primal-dual Newton Conjugate Gradients method for image denoising and deblurring problems. In this paper we modify their method and adapt it for CS problems. There are two major contributions.

First, we propose an inexpensive preconditioner for fast solution of systems using pdNCG when applied to CS problems with coherent and redundant dictionaries. The proposed preconditioner is a generalization of the preconditioner in [16] for CS problems with incoherent dictionaries. We analyze the limiting behaviour of our preconditioner and prove that the eigenvalues of the preconditioned matrices are clustered around one. This is an essential property that guarantees that only a few iterations of CG will be needed to approximately solve the linear systems. Moreover, we provide computational evidence that the preconditioner works well not only close to the solution (as predicted by its spectral analysis) but also in earlier iterations of pdNCG.

Second, we demonstrate that despite being a second-order method, pdNCG can be more efficient than specialized first-order methods for CS problems of our interest, even on large-scale instances. This performance is observed in several numerical experiments presented in this paper. We believe that the reason for this is that pdNCG, as a second-order method, captures the curvature of the problems, which results in sufficient decrease in the number of iterations compared to first-order methods. This advantage comes with the computational cost of having to solve a linear system at every iteration. However, inexact solution of the linear systems using CG combined with the proposed efficient preconditioner crucially reduces the computational costs per iteration.

1.4. Format of the paper and notation. The paper is organized as follows. In Section 2, problem (1.1) is replaced by a smooth approximation; the ℓ_1 -norm is approximated by the pseudo-Huber function. Derivation of pseudo-Huber function is discussed and its derivatives are calculated. In Section 3, a primal-dual reformulation of the approximation to problem (1.1) and its optimality conditions are obtained. In Section 4, pdNCG is presented. For convergence analysis of pdNCG method the reader is referred to [11, 15]. In Section 5, a preconditioning technique is described for controlling the spectrum of matrices in systems which arise. In Section 6, a continuation framework for pdNCG is described. In Section 7, numerical experiments are discussed that present the efficiency of pdNCG. Finally, in Section 8, conclusions are made.

Throughout the paper, $\|\cdot\|_1$ is the ℓ_1 -norm, $\|\cdot\|_2$ is the Euclidean norm, $\|\cdot\|_\infty$ the infinity norm and $|\cdot|$ is the absolute value. The functions $Re(\cdot)$ and $Im(\cdot)$ take a complex input and return its real and imaginary part, respectively. For simplification of notation, occasionally we will use $Re(\cdot)$ and $Im(\cdot)$ without the parenthesis. Furthermore, $diag(\cdot)$ denotes the function which takes as input a vector and outputs a

diagonal square matrix with the vector in the main diagonal. Finally, the super index c denotes the complementarity set, i.e. \mathcal{B}^c is the complement set of \mathcal{B} .

2. Regularization by pseudo-Huber. In pdNCG [11] the non-differentiability of the ℓ_1 -norm is treated by applying smoothing. In particular, the ℓ_1 -norm is replaced with the pseudo-Huber function [18]

$$(2.1) \quad \psi_\mu(W^*x) := \sum_{i=1}^l ((\mu^2 + |W_i^*x|^2)^{\frac{1}{2}} - \mu),$$

where W_i is the i^{th} column of matrix $W \in E^{n \times l}$ and μ controls the quality of approximation, i.e. for $\mu \rightarrow 0$, $\psi_\mu(x)$ tends to the ℓ_1 -norm. The original problem (1.1) is approximated by

$$(2.2) \quad \text{minimize } f_c^\mu(x) := c\psi_\mu(W^*x) + \frac{1}{2}\|Ax - b\|_2^2.$$

2.1. Derivation of pseudo-Huber function. The pseudo-Huber function (2.1) can be derived in a few simple steps. First, we re-write function $\|W^*x\|_1$ in its dual form

$$(2.3) \quad \|W^*x\|_1 = \sup_{g \in \mathbb{C}^l, \|g\|_\infty \leq 1} \text{Re}(\bar{g}^* W^*x),$$

where g are dual variables. The pseudo-Huber function is obtained by regularizing the previous dual form

$$(2.4) \quad \psi_\mu(W^*x) = \sup_{g \in \mathbb{C}^l, \|g\|_\infty \leq 1} \text{Re}(\bar{g}^* W^*x) + \sum_{i=1}^l \left(\mu(1 - |g_i|^2)^{\frac{1}{2}} - \mu \right),$$

where g_i is the i^{th} component of vector g .

An approach which consists of smoothing a non-smooth function through its dual form is known as Moreau's proximal smoothing technique [22]. Another way to smooth function $\|W^*x\|_1$ is to regularize its dual form with a strongly convex quadratic function $\mu/2\|g\|_2^2$. Such an approach provides a smooth approximation of $\|W^*x\|_1$ which is known as Huber function and it has been used in [4]. Generalizations of the Moreau proximal smoothing technique can be found in [24] and [3].

2.2. Derivatives of pseudo-Huber function. The gradient of pseudo-Huber function $\psi_\mu(W^*x)$ in (2.1) is given by

$$\nabla \psi_\mu(W^*x) = \text{Re}(W D W^*)x,$$

where $D := \text{diag}(D_1, D_2, \dots, D_l)$ with

$$(2.5) \quad D_i := (\mu^2 + |y_i|^2)^{-\frac{1}{2}} \quad \forall i = 1, 2, \dots, l,$$

and $y = [y_1, y_2, \dots, y_l]^T := W^*x$. The gradient of function $f_c^\mu(x)$ in (2.2) is

$$\nabla f_c^\mu(x) = c\nabla \psi_\mu(W^*x) + A^T(Ax - b).$$

The Hessian matrix of $\psi_\mu(x)$ is

$$(2.6) \quad \nabla^2 \psi_\mu(W^*x) := \frac{1}{4}(W \hat{Y} W^* + \bar{W} \hat{Y} \bar{W}^* + W \tilde{Y} \bar{W}^* + \bar{W} \tilde{Y} W^*),$$

where the bar symbol denotes the complex conjugate, $\hat{Y} := \text{diag} [\hat{Y}_1, \hat{Y}_2, \dots, \hat{Y}_l]$, $\tilde{Y} := \text{diag} [\tilde{Y}_1, \tilde{Y}_2, \dots, \tilde{Y}_l]$ and

$$(2.7) \quad \hat{Y}_i := \mu^2 D_i^3 + D_i, \quad \tilde{Y}_i := -y_i^2 D_i^3, \quad i = 1, 2, \dots, l,$$

Moreover, the Hessian matrix of $f_c^\mu(x)$ is

$$(2.8) \quad \nabla^2 f_c^\mu(x) = c \nabla^2 \psi_\mu(W^*x) + A^\top A.$$

3. Primal-dual formulation and optimality conditions. In [10] the authors solved iTV problems for square and full-rank matrices A which were inexpensively diagonalizable, i.e. image deblurring or denoising. More precisely, in the previous cited paper the authors tackled iTV problems using a Newton-CG method to solve problem (2.2). They observed that close to the points of non-smoothness of the ℓ_1 -norm, the smooth pseudo-Huber function (2.1) exhibited an ill-conditioning behaviour. This results in two major drawbacks of the application of Newton-CG. First, the linear algebra is challenging. Second, the region of convergence of Newton-CG is substantially shrunk. To deal with these problems they proposed to incorporate Newton-CG inside a continuation procedure on the parameters c and μ . Although they showed that continuation did improve the global convergence properties of Newton-CG it was later discussed in [11] (for the same iTV problems) that continuation was difficult to control (especially for small μ) and Newton-CG was not always convergent in reasonable CPU time.

In [11], Chan, Golub and Mulet provided numerical evidence that the behaviour of a Newton-CG method can be made significantly more robust even for small values of μ . This is achieved by simply solving a primal-dual reformulation of (2.2), which is given below

$$(3.1) \quad \text{minimize} \quad \sup_{g \in \mathbb{C}^l, \|g\|_\infty \leq 1} c \text{Re}(\bar{g}^* W^*)x + c \sum_{i=1}^l \left(\mu(1 - |g_i|^2)^{1/2} - \mu \right) + \frac{1}{2} \|Ax - b\|_2^2.$$

The reason that Newton-CG method is more robust when applied on problem (3.1) than on problem (2.2) is hidden in the linearization of the optimality conditions of the two problems.

3.1. Optimality conditions. The optimality conditions of problem (2.2) are

$$(3.2) \quad \nabla \psi_\mu(W^*x) + A^\top(Ax - b) = c \text{Re}(W D W^*)x + A^\top(Ax - b) = 0.$$

The first-order optimality conditions of the primal-dual problem (3.1) are

$$(3.3) \quad \begin{aligned} c \text{Re}(W \bar{g}) + A^\top(Ax - b) &= 0, \\ D^{-1} \bar{g} &= W^*x. \end{aligned}$$

Notice for conditions (3.3) that the constraint $\|g\|_\infty \leq 1$ in (3.1) is redundant since any x and g that satisfy (3.3) also satisfy this constraint. Hence, the constraint has been dropped. Conditions (3.3) are obtained from (3.2) by simply setting $\bar{g} = D W^*x$. Hence, their only difference is the inversion of matrix D . However, this small difference affects crucially the performance of Newton-CG.

The reason behind this is that the linearization of the second equation in (3.3), i.e. $D^{-1} \bar{g} = W^*x$, is of much better quality than the linearization of $\nabla \psi_\mu(W^*x)$ for

$\mu \approx 0$ and $W^*x \approx 0$. To see why this is true, observe that for small μ and $W^*x \approx 0$, the gradient $\nabla\psi_\mu(W^*x)$ becomes close to singular and its linearization is expected to be inaccurate. On the other hand, $D^{-1}\bar{g} = W^*x$ as a function of W^*x is not singular for $\mu \approx 0$ and $W^*x \approx 0$, hence, its linearization is expected to be more accurate. We refer the reader to Section 3 of [11] for empirical justification.

4. Primal-dual Newton conjugate gradients method. In this section we present details of pdNCG method [11].

4.1. The method. First, we convert optimality conditions (3.3) to the real case. This is done by splitting matrix $W = \text{Re}W + \sqrt{-1}\text{Im}W$ and the dual variables $g = g_{re} + \sqrt{-1}g_{im}$ into their real and imaginary parts. We do this in order to obtain optimality conditions which are differentiable in the classical sense of real analysis. This allows a straightforward application of pdNCG method [11]. The real optimality conditions of the primal-dual problem (3.1) are given below

$$(4.1) \quad \begin{aligned} c(\text{Re}Wg_{re} + \text{Im}Wg_{im}) + A^\top(Ax - b) &= 0, \\ D^{-1}g_{re} &= \text{Re}W^\top x, \quad D^{-1}g_{im} = \text{Im}W^\top x. \end{aligned}$$

At every iteration of pdNCG the primal-dual directions are calculated by approximate solving the following linearization of the equality constraints in (4.1)

$$(4.2) \quad \begin{aligned} B\Delta x &= -\nabla f_c^\mu(x) \\ \Delta g_{re} &= D(I - B_1)\text{Re}W^\top \Delta x + DB_2\text{Im}W^\top \Delta x - g_{re} + D\text{Re}W^\top x \\ \Delta g_{im} &= D(I - B_4)\text{Im}W^\top \Delta x + DB_3\text{Re}W^\top \Delta x - g_{im} + D\text{Im}W^\top x \end{aligned}$$

where

$$(4.3) \quad B := c\tilde{B} + A^\top A,$$

$$\begin{aligned} \tilde{B} &:= \text{Re}WD(I - B_1)\text{Re}W^\top + \text{Im}WD(I - B_4)\text{Im}W^\top + \text{Re}WDB_2\text{Im}W^\top \\ &\quad + \text{Im}WB_3D\text{Re}W^\top, \end{aligned}$$

and $B_i, i = 1, 2, 3, 4$ are diagonal matrices with components

$$\begin{aligned} [B_1]_{ii} &:= D_i[g_{re}]_i \text{Re}W_i^\top x, & [B_2]_{ii} &:= D_i[g_{re}]_i \text{Im}W_i^\top x, \\ [B_3]_{ii} &:= D_i[g_{im}]_i \text{Re}W_i^\top x, & [B_4]_{ii} &:= D_i[g_{im}]_i \text{Im}W_i^\top x. \end{aligned}$$

REMARK 4.1. *Matrix B in (4.3) is positive definite (and invertible) if $\|g_{re} + \sqrt{-1}g_{im}\|_\infty \leq 1$ and condition (1.4) are satisfied. The former condition will be maintained through all iterations of pdNCG.*

It is straightforward to show the claim in Remark 4.1 for the case of W being a real matrix. For the case of complex W we refer the reader to a similar claim which is made in [11], page 1970. Although matrix B is positive definite under the conditions stated in Remark 4.1, it is not symmetric, except in the case that W is real where all imaginary parts are dropped. Therefore in the case of complex matrix W , preconditioned CG (PCG) cannot be employed to approximately solve (4.2). To avoid the problem of non-symmetric matrix B the authors in [11] suggested to ignore the non-symmetric part in matrix B and employ CG to solve (4.2). This idea is based on the following remark.

- 1: **Input:** $\tau_1 \in (0, 1)$, $\tau_2 \in (0, 1/2)$, x^0 , g_{re}^0 and g_{im}^0 , where $\|g_{re}^0 + \sqrt{-1}g_{im}^0\|_\infty \leq 1$.
- 2: **Loop:** For $k = 1, 2, \dots$, until termination criteria are met.
- 3: Calculate Δx^k , Δg_{re}^k and Δg_{im}^k by solving approximately the system (4.4), until (4.6) is satisfied for some $\eta \in [0, 1)$.
- 4: Set $\tilde{g}_{re}^{k+1} := g_{re}^k + \Delta g_{re}^k$, $\tilde{g}_{im}^{k+1} := g_{im}^k + \Delta g_{im}^k$ and calculate

$$\bar{g}^{k+1} := P_{\|\cdot\|_\infty \leq 1}(\tilde{g}_{re}^{k+1} + \sqrt{-1}\tilde{g}_{im}^{k+1}),$$

where $P_{\|\cdot\|_\infty \leq 1}(\cdot)$ is the orthogonal projection on the ℓ_∞ ball.

Then set $g_{re}^{k+1} := \text{Re}\bar{g}^{k+1}$ and $g_{im}^{k+1} := \text{Im}\bar{g}^{k+1}$.

- 5: Find the least integer $j \geq 0$ such that

$$f_c^\mu(x^k + \tau_1^j \Delta x^k) \leq f_c^\mu(x^k) + \tau_2 \tau_1^j (\nabla f_c^\mu(x^k))^\top \Delta x^k$$

and set $\alpha := \tau_1^j$.

- 6: Set $x^{k+1} := x^k + \alpha \Delta x^k$.

Fig. 4.1: Algorithm primal-dual Newton Conjugate Gradients

REMARK 4.2. *The symmetric part of B tends to the symmetric second-order derivative of $f_c^\mu(x)$ as pdNCG converges (see Section 5 in [11]).*

Hence, system (4.2) is replaced with

$$(4.4) \quad \begin{aligned} \hat{B} \Delta x &= -\nabla f_c^\mu(x) \\ \Delta g_{re} &= D(I - B_1) \text{Re} W^\top \Delta x + D B_2 \text{Im} W^\top \Delta x - g_{re} + D \text{Re} W^\top x \\ \Delta g_{im} &= D(I - B_4) \text{Im} W^\top \Delta x + D B_3 \text{Re} W^\top \Delta x - g_{im} + D \text{Im} W^\top x \end{aligned}$$

where

$$(4.5) \quad \hat{B} := c \text{sym}(\tilde{B}) + A^\top A$$

and $\text{sym}(\tilde{B}) := 1/2(\tilde{B} + \tilde{B}^\top)$ is the symmetric part of \tilde{B} . Moreover, PCG is terminated when

$$(4.6) \quad \|\hat{B} \Delta x + \nabla f_c^\mu(x)\|_2 \leq \eta \|\nabla f_c^\mu(x)\|_2,$$

is satisfied for $\eta \in [0, 1)$. Then the iterate $g = g_{re} + \Delta g_{re} + \sqrt{-1}(g_{im} + \Delta g_{im})$ is orthogonally projected on the box $\{x : \|x\|_\infty \leq 1\}$. The projection operator for complex arguments is applied component-wise and it is defined as $v := P_{\|\cdot\|_\infty \leq 1}(u) = \min(1/|u|, 1) \odot u$, where \odot denotes the component-wise multiplication. In the last step, line-search is employed for the primal Δx direction in order to guarantee that the objective value $f_c^\mu(x)$ is monotonically decreasing, see Section 5 of [11]. The pseudo-code of pdNCG is presented in Figure 4.1.

5. Preconditioning. Practical computational efficiency of pdNCG applied to system (4.4) depends on spectral properties of matrix \hat{B} in (4.5). Those can be improved by a suitable preconditioning. In this section we introduce a new preconditioner for \hat{B} and discuss the limiting behaviour of the spectrum of preconditioned \hat{B} .

First, we give an intuitive analysis on the construction of the proposed preconditioner. In Remark A.2 it is mentioned that the distance ω of the two solutions $x_c := \arg \min f_c(x)$ and $x_{c,\mu} := \arg \min f_c^\mu(x)$ can be arbitrarily small for sufficiently small values of μ . Moreover, according to Assumption 1.2, W^*x_c is q sparse. Therefore, Remark A.2 implies that $W^*x_{c,\mu}$ is approximately q sparse with nearly zero components of $\mathcal{O}(\omega)$. A consequence of the previous statement is that the components of $W^*x_{c,\mu}$ split into the following disjoint sets

$$\mathcal{B} := \{i \in \{1, 2, \dots, l\} \mid |W_i^*x_{c,\mu}| \gg \mathcal{O}(\omega)\}, \quad |\mathcal{B}| = q = |\text{supp}(W^*x_c)|,$$

$$\mathcal{B}^c := \{i \in \{1, 2, \dots, l\} \mid |W_i^*x_{c,\mu}| \approx \mathcal{O}(\omega)\}, \quad |\mathcal{B}^c| = l - q.$$

The behaviour of $W^*x_{c,\mu}$ has a crucial influence on matrix $\nabla^2\psi_\mu(W^*x_{c,\mu})$ in (2.6). Notice that the components of the diagonal matrix D , defined in (2.5) as part of $\nabla^2\psi_\mu(W^*x_{c,\mu})$, split into two disjoint sets. In particular, q components are non-zeros much less than $\mathcal{O}(1/\omega)$, while the majority, $l - q$, of its components are of $\mathcal{O}(1/\omega)$,

$$(5.1) \quad D_i \ll \mathcal{O}\left(\frac{1}{\omega}\right) \quad \forall i \in \mathcal{B} \quad \text{and} \quad D_i = \mathcal{O}\left(\frac{1}{\omega}\right) \quad \forall i \in \mathcal{B}^c.$$

Hence, for points close to $x_{c,\mu}$ and small μ , matrix $\nabla^2 f_c^\mu(x)$ in (2.8) consists of a dominant matrix $c\nabla^2\psi_\mu(x)$ and of matrix $A^\top A$ with moderate largest eigenvalue. The previous argument for $A^\top A$ is due to (1.3). Observe that $\lambda_{\max}(A^\top A) = \lambda_{\max}(AA^\top)$, hence, if δ in (1.3) is not a very large constant, then $\lambda_{\max}(A^\top A) \leq 1 + \delta$. According to Remark 4.2, the symmetric matrix $\text{sym}(\tilde{B})$ in (2.8) tends to matrix $\nabla^2\psi_\mu(x)$ as $x \rightarrow x_{c,\mu}$. Therefore, matrix $\text{sym}(\tilde{B})$ is the dominant matrix in \hat{B} . For this reason, in the proposed preconditioning technique, matrix $A^\top A$ in (2.8) is replaced by a scaled identity ρI_n , $\rho > 0$, while the dominant matrix $\text{sym}(\tilde{B})$ is maintained. Based on these observations we propose the following preconditioner

$$(5.2) \quad \tilde{N} := c\text{sym}(\tilde{B}) + \rho I_n.$$

In order to capture the approximate separability of the diagonal components of matrix D for points close to $x_{c,\mu}$, when μ is sufficiently small, we will work with approximate guess of \mathcal{B} and \mathcal{B}^c . For this reason, we introduce the positive constant ν , such that

$$\#(D_i < \nu) = \sigma.$$

Here σ might be different from the sparsity of W^*x_c . Furthermore, according to the above definition we have the sets

$$(5.3) \quad \mathcal{B}_\nu := \{i \in \{1, 2, \dots, l\} \mid D_i < \nu\} \quad \text{and} \quad \mathcal{B}_\nu^c := \{1, 2, \dots, l\} \setminus \mathcal{B}_\nu,$$

with $|\mathcal{B}_\nu| = \sigma$ and $|\mathcal{B}_\nu^c| = l - \sigma$. This notation is being used in the following theorem, in which we analyze the behaviour of the spectral properties of preconditioned $\nabla^2 f_c^\mu(x)$, with preconditioner $N := c\nabla^2\psi_\mu(W^*x) + \rho I_n$. However, according to Remark 4.2 matrices \hat{B} and \tilde{N} tend to $\nabla^2 f_c^\mu(x)$ and N , respectively, as $x \rightarrow x_{c,\mu}$. Therefore, the following theorem is useful for the analysis of the limiting behaviour of the spectrum of preconditioned \hat{B} .

THEOREM 5.1. Let ν be any positive constant and $\#(D_i < \nu) = \sigma$ at a point x , where D is defined in (2.5). Let

$$\nabla^2 f_c^\mu(x) = c\nabla^2 \psi_\mu(W^*x) + A^\top A \quad \text{and} \quad N := c\nabla^2 \psi_\mu(W^*x) + \rho I_n.$$

Additionally, let A and W satisfy W -RIP with some constant $\delta_\sigma < 1/2$ and let A satisfy (1.3) for some constant $\delta \geq 0$.

If the eigenvectors of $N^{-\frac{1}{2}} \nabla^2 f_c^\mu(x) N^{-\frac{1}{2}}$ do not belong in $\text{Ker}(W_{\mathcal{B}_\nu^c}^*)$ and $\rho \in [\delta_\sigma, 1/2]$, then the eigenvalues of $N^{-1} \nabla^2 f_c^\mu(x)$ satisfy

$$|\lambda - 1| \leq \frac{1}{2} \frac{\chi + 1 + (5\chi^2 - 2\chi + 1)^{\frac{1}{2}}}{c\mu^2\nu^3\lambda_{\min}(\text{Re}(W_{\mathcal{B}_\nu^c}W_{\mathcal{B}_\nu^c}^*)) + \rho},$$

where $\lambda \in \text{spec}(N^{-1} \nabla^2 f_c^\mu(x))$, $\lambda_{\min}(\text{Re}(W_{\mathcal{B}_\nu^c}W_{\mathcal{B}_\nu^c}^*))$ is the minimum nonzero eigenvalue of $\text{Re}(W_{\mathcal{B}_\nu^c}W_{\mathcal{B}_\nu^c}^*)$ and $\chi := 1 + \delta - \rho$.

If the eigenvectors of $N^{-\frac{1}{2}} \nabla^2 f_c^\mu(x) N^{-\frac{1}{2}}$ belong in $\text{Ker}(W_{\mathcal{B}_\nu^c}^*)$, then

$$|\lambda - 1| \leq \frac{1}{2} \frac{\chi + 1 + (5\chi^2 - 2\chi + 1)^{\frac{1}{2}}}{\rho}.$$

Proof. We analyze the spectrum of matrix $N^{-\frac{1}{2}} \nabla^2 f_c^\mu(x) N^{-\frac{1}{2}}$ instead, because it has the same eigenvalues as matrix $N^{-1} \nabla^2 f_c^\mu(x)$. We have that

$$\begin{aligned} N^{-\frac{1}{2}} \nabla^2 f_c^\mu(x) N^{-\frac{1}{2}} &= N^{-\frac{1}{2}} (c\nabla^2 \psi_\mu(x) + A^\top A) N^{-\frac{1}{2}} \\ &= N^{-\frac{1}{2}} (c\nabla^2 \psi_\mu(x) + A^\top A + \rho I_n - \rho I_n) N^{-\frac{1}{2}} \\ &= N^{-\frac{1}{2}} (c\nabla^2 \psi_\mu(x) + \rho I_n) N^{-\frac{1}{2}} + N^{-\frac{1}{2}} A^\top A N^{-\frac{1}{2}} - \rho N^{-1} \\ &= I_n + N^{-\frac{1}{2}} A^\top A N^{-\frac{1}{2}} - \rho N^{-1} \end{aligned}$$

Let u be an eigenvector of $N^{-\frac{1}{2}} \nabla^2 f_c^\mu(x) N^{-\frac{1}{2}}$ with $\|u\|_2 = 1$ and λ the corresponding eigenvalue, then

$$\begin{aligned} (I_n + N^{-\frac{1}{2}} A^\top A N^{-\frac{1}{2}} - \rho N^{-1})u &= \lambda u && \Longleftrightarrow \\ (N + N^{\frac{1}{2}} A^\top A N^{-\frac{1}{2}} - \rho I_n)u &= \lambda Nu && \Longrightarrow \\ u^\top N^{\frac{1}{2}} (A^\top A - \rho I_n) N^{-\frac{1}{2}} u &= (\lambda - 1)u^\top Nu && \Longrightarrow \\ (5.4) \quad |u^\top N^{\frac{1}{2}} (A^\top A - \rho I_n) N^{-\frac{1}{2}} u| &= |\lambda - 1|u^\top Nu. \end{aligned}$$

First, we find an upper bound for $|u^\top N^{\frac{1}{2}} (A^\top A - \rho I_n) N^{-\frac{1}{2}} u|$. Matrices $N^{\frac{1}{2}} (A^\top A - \rho I_n) N^{-\frac{1}{2}}$ and $A^\top A - \rho I_n$ have the same eigenvalues. Therefore,

$$|u^\top N^{\frac{1}{2}} (A^\top A - \rho I_n) N^{-\frac{1}{2}} u| \leq \lambda_{\max}^+(A^\top A - \rho I_n)$$

where $\lambda_{\max}^+(\cdot)$ is the largest eigenvalue of the input matrix in absolute value. Thus,

$$\begin{aligned} |u^\top N^{\frac{1}{2}} (A^\top A - \rho I_n) N^{-\frac{1}{2}} u| &\leq \max_{\|v\|_2^2 \leq 1} |v^\top (A^\top A - \rho I_n) v| \\ &= \max_{\|Pv\|_2^2 + \|Qv\|_2^2 \leq 1} |(Pv + Qv)^\top (A^\top A - \rho I_n) (Pv + Qv)|, \end{aligned}$$

where P is the projection matrix to the column space of $W_{\mathcal{B}_\nu}$ and $Q = I_n - P$. Using triangular inequality we get

$$\begin{aligned} |u^\top N^{\frac{1}{2}}(A^\top A - \rho I_n)N^{-\frac{1}{2}}u| &\leq \max_{\|Pv\|_2^2 + \|Qv\|_2^2 \leq 1} (|(Pv)^\top (A^\top A - \rho I_n)Pv| \\ &\quad + |(Qv)^\top (A^\top A - \rho I_n)Qv| + 2|(Pv)^\top (A^\top A - \rho I_n)Qv|). \end{aligned}$$

Let us denote by \hat{v} the solution of this maximization problem and set $\|P\hat{v}\|_2^2 = \alpha$ and $\|Q\hat{v}\|_2^2 = 1 - \alpha$, where $\alpha \in [0, 1]$, then

$$(5.5) \quad \begin{aligned} |u^\top N^{\frac{1}{2}}(A^\top A - \rho I_n)N^{-\frac{1}{2}}u| &\leq (|(P\hat{v})^\top (A^\top A - \rho I_n)P\hat{v}| \\ &\quad + |(Q\hat{v})^\top (A^\top A - \rho I_n)Q\hat{v}| + 2|(P\hat{v})^\top (A^\top A - \rho I_n)Q\hat{v}|). \end{aligned}$$

Since $P\hat{v}$ belongs to the column space of $W_{\mathcal{B}_\nu}$ and $|\mathcal{B}_\nu| = \sigma$, from W-RIP with $\delta_\sigma < 1/2$ we have that

$$\begin{aligned} \|P\hat{v}\|_2^2(1 - \delta_\sigma) &\leq \|AP\hat{v}\|_2^2 && \implies \\ \|P\hat{v}\|_2^2(1 - \rho) &\leq \|AP\hat{v}\|_2^2 && \iff \\ \|P\hat{v}\|_2^2(1 - 2\rho) &\leq \|AP\hat{v}\|_2^2 - \rho\|P\hat{v}\|_2^2. \end{aligned}$$

Since $\rho \in [\delta_\sigma, 1/2]$ we have that $\rho\|P\hat{v}\|_2^2 \leq \|AP\hat{v}\|_2^2$, which implies that if the eigenvector corresponding to an eigenvalue of matrix $A^\top A$ belongs to the column space of $W_{\mathcal{B}_\nu}$, then the eigenvalue cannot be smaller than ρ . Hence,

$$|(P\hat{v})^\top (A^\top A - \rho I_n)P\hat{v}| \leq |(P\hat{v})^*(A^\top A - \rho I_n)P\hat{v}| = (P\hat{v})^*(A^\top A - \rho I_n)P\hat{v}.$$

Moreover, from W-RIP with $\delta_\sigma < 1/2$ and $\rho \in [\delta_\sigma, 1/2]$, we also have that $(P\hat{v})^*(A^\top A - \rho I_n)P\hat{v} \leq \|P\hat{v}\|_2^2$. Thus,

$$(5.6) \quad |(P\hat{v})^\top (A^\top A - \rho I_n)P\hat{v}| \leq \alpha.$$

From property (1.3) and $\lambda_{\max}(A^\top A) = \lambda_{\max}(AA^\top)$, we have that $\lambda_{\max}(A^\top A - \rho I_n) \leq 1 + \delta - \rho$. Finally, using the Cauchy-Schwarz inequality, we get that

$$(5.7) \quad |(Q\hat{v})^\top (A^\top A - \rho I_n)Q\hat{v}| \leq (1 + \delta - \rho)(1 - \alpha)$$

and

$$(5.8) \quad |(P\hat{v})^\top (A^\top A - \rho I_n)Q\hat{v}| \leq (1 + \delta - \rho)\sqrt{\alpha(1 - \alpha)}.$$

Using (5.6), (5.7) and (5.8) in (5.5) we have that

$$(5.9) \quad |u^\top N^{\frac{1}{2}}(A^\top A - \rho I_n)N^{-\frac{1}{2}}u| \leq \alpha + (1 + \delta - \rho)(1 - \alpha) + 2(1 + \delta - \rho)\sqrt{\alpha(1 - \alpha)}.$$

Set $\chi := 1 + \delta - \rho$, it is easy to check that in the interval $\alpha \in [0, 1]$ the right hand side of (5.9) has a maximum at one of the four candidate points

$$\alpha_1 = 0, \quad \alpha_2 = 1, \quad \alpha_{3,4} = \frac{1}{2} \left(1 \pm \left(\frac{(\chi - 1)^2}{5\chi^2 - 2\chi + 1} \right)^{1/2} \right),$$

where α_3 is for plus and α_4 is for minus. The corresponding function values are

$$\chi, \quad 1, \quad \frac{\chi + 1}{2} + \frac{1}{2} \frac{3\chi^2 + 2\chi - 1}{(5\chi^2 - 2\chi + 1)^{1/2}}, \quad \frac{\chi + 1}{2} + \frac{1}{2} (5\chi^2 - 2\chi + 1)^{1/2},$$

respectively. Hence, the maximum among these four values is given for α_4 . Thus, (5.9) is upper bounded by

$$(5.10) \quad |u^\top N^{\frac{1}{2}}(A^\top A - \rho I_n)N^{-\frac{1}{2}}u| \leq \frac{\chi + 1}{2} + \frac{1}{2}(5\chi^2 - 2\chi + 1)^{\frac{1}{2}}.$$

We now find a lower bound for $u^\top Nu$. Using the definition of D in (2.5), matrix \hat{Y} in (2.7) is rewritten as $\hat{Y}_i = (2\mu^2 + |y_i|^2)D_i^3 \forall i = 1, 2, \dots, l$. Thus $\nabla^2 \psi_\mu(x)$ in (2.6) is rewritten as

$$(5.11) \quad \begin{aligned} \nabla^2 \psi_\mu(W^*x) &= \frac{1}{4}[(W\tilde{D}^3W^* + \bar{W}\tilde{D}^3\bar{W}^* + W\tilde{Y}\bar{W}^* + \bar{W}\tilde{Y}W^*) \\ &\quad + 2\mu^2(WD^3W^* + \bar{W}D^3\bar{W}^*)], \end{aligned}$$

where $\tilde{D}_i = |y_i|^2 D_i^3 \forall i = 1, 2, \dots, l$. Observe, that matrix $\nabla^2 \psi_\mu(W^*x)$ consists of two matrices $W\tilde{D}^3W^* + \bar{W}\tilde{D}^3\bar{W}^* + W\tilde{Y}\bar{W}^* + \bar{W}\tilde{Y}W^*$ and $2\mu^2(WD^3W^* + \bar{W}D^3\bar{W}^*)$ which are positive semi-definite. Using (5.11) and the previous statement we get that

$$\begin{aligned} u^\top Nu &= u^\top (c\nabla^2 \psi_\mu(W^*x) + \rho I_n)u \\ &= \frac{c}{4}u^\top (W\hat{Y}W^* + \bar{W}\hat{Y}\bar{W}^* + W\tilde{Y}\bar{W}^* + \bar{W}\tilde{Y}W^*)u + \rho \\ &\geq \frac{c\mu^2}{2}u^\top (WD^3W^* + \bar{W}D^3\bar{W}^*)u + \rho. \end{aligned}$$

Furthermore, using the splitting of matrix D (5.3), the last inequality is equivalent to

$$\begin{aligned} u^\top Nu &= \frac{c\mu^2}{2}u^\top (W_{\mathcal{B}_\nu} D_{\mathcal{B}_\nu}^3 W_{\mathcal{B}_\nu}^* + W_{\mathcal{B}_\nu^c} D_{\mathcal{B}_\nu^c}^3 W_{\mathcal{B}_\nu^c}^* + \bar{W}_{\mathcal{B}_\nu} D_{\mathcal{B}_\nu}^3 \bar{W}_{\mathcal{B}_\nu}^* + \bar{W}_{\mathcal{B}_\nu^c} D_{\mathcal{B}_\nu^c}^3 \bar{W}_{\mathcal{B}_\nu^c}^*)u + \rho \\ &\geq \frac{c\mu^2}{2}u^\top (W_{\mathcal{B}_\nu^c} D_{\mathcal{B}_\nu^c}^3 W_{\mathcal{B}_\nu^c}^* + \bar{W}_{\mathcal{B}_\nu^c} D_{\mathcal{B}_\nu^c}^3 \bar{W}_{\mathcal{B}_\nu^c}^*)u + \rho. \end{aligned}$$

Using the defition of \mathcal{B}_ν^c (5.3) in the last inequality, the quantity $u^\top Nu$ is further lower bounded by

$$(5.12) \quad u^\top Nu \geq \frac{c\mu^2\nu^3}{2}u^\top (W_{\mathcal{B}_\nu^c} W_{\mathcal{B}_\nu^c}^* + \bar{W}_{\mathcal{B}_\nu^c} \bar{W}_{\mathcal{B}_\nu^c}^*)u + \rho.$$

If $u \notin \text{Ker}(W_{\mathcal{B}_\nu^c}^*)$, then from (5.12) we get

$$(5.13) \quad u^\top Nu \geq c\mu^2\nu^3\lambda_{\min}(\text{Re}(W_{\mathcal{B}_\nu^c} W_{\mathcal{B}_\nu^c}^*)) + \rho.$$

Hence, combining (5.4), (5.10) and (5.13) we conclude that

$$|\lambda - 1| \leq \frac{1}{2} \frac{\chi + 1 + (5\chi^2 - 2\chi + 1)^{\frac{1}{2}}}{c\mu^2\nu^3\lambda_{\min}(\text{Re}(W_{\mathcal{B}_\nu^c} W_{\mathcal{B}_\nu^c}^*)) + \rho}$$

If $u \in \text{Ker}(W_{\mathcal{B}_\nu^c}^*)$, then from (5.12) we have that $u^\top Nu \geq \rho$, hence

$$|\lambda - 1| \leq \frac{1}{2} \frac{\chi + 1 + (5\chi^2 - 2\chi + 1)^{\frac{1}{2}}}{\rho}.$$

□

Let us now draw some conclusions from Theorem 5.1. In order for the eigenvalues of $N^{-1}\nabla^2 f_c^\mu(x)$ to be around one, it is required that the degree of freedom ν is chosen such that $\nu = \mathcal{O}(1/\mu)$ and μ is small. For such ν , the cardinality σ of the set \mathcal{B}_ν must be small enough such that matrices A and W satisfy W-RIP with constant $\delta_\sigma < 1/2$; otherwise the assumptions of Theorem 5.1 will not be satisfied. This is possible if the pdNCG iterates are close to the optimal solution $x_{c,\mu}$ and μ is sufficiently small. In particular, for sufficiently small μ , from Remark A.2 we have that $x_{c,\mu} \approx x_c$ and $\sigma \approx q$. According to Assumption 1.2 for the q -sparse x_c , W-RIP is satisfied for $\delta_{2q} < 1/2 \implies \delta_q < 1/2$. Hence, for points close to $x_{c,\mu}$ and small μ we expect that $\delta_\sigma < 1/2$. Therefore, the result in Theorem 5.1 captures only the limiting behaviour of preconditioned $\nabla^2 f_c^\mu(x)$ as $x \rightarrow x_{c,\mu}$. Moreover, according to Remark 4.2, Theorem 5.1 implies that at the limit the eigenvalues of $\tilde{N}^{-1}\hat{B}$ are also clustered around one.

We now comment on the second result of Theorem 5.1, when the eigenvectors of $N^{-\frac{1}{2}}\nabla^2 f_c^\mu(x)N^{-\frac{1}{2}}$ belong in $\text{Ker}(W_{\mathcal{B}_\nu}^*)$. In this case, according to Theorem 5.1 the preconditioner removes the disadvantageous dependence of the spectrum of $\nabla^2 f_c^\mu(x)$ on the smoothing parameter μ . However, there is no guarantee that the eigenvalues of $N^{-1}\nabla^2 f_c^\mu(x)$ are clustered around one, regardless of the distance from the optimal solution $x_{c,\mu}$. Again, because of Remark 4.2 we expect that the spectrum of $\tilde{N}^{-1}\hat{B}$ at the limit will have a similar behaviour.

The scenario of limiting behaviour of the preconditioner is pessimistic. Let $\tilde{\sigma}$ be the minimum sparsity level such that matrices A and W are W-RIP with $\delta_{\tilde{\sigma}} < 1/2$. Then, according to the uniform property of W-RIP (i.e. it holds for all at most $\tilde{\sigma}$ -sparse vectors), the preconditioner will start to be effective even if the iterates W^*x^k are approximately sparse with $\tilde{\sigma}$ dominant non-zero components. Numerical evidence is provided in Figure 5.1 to confirm this claim. In Figure 5.1 the spectra $\lambda(\hat{B})$ and $\lambda(\tilde{N}^{-1}\hat{B})$ are displayed for a sequence of systems which arise when an iTV problem is solved. For this iTV problem we set matrix A to be a partial 2D DCT, $n = 2^{10}$, $m = n/4$, $c = 2.29e-2$ and $\rho = 5.0e-1$. For the first experiment, which corresponds to Figures 5.1a and 5.1c the smoothing parameter has been set to $\mu = 1.0e-3$. For the second experiment, which corresponds to Figures 5.1b and 5.1d we set $\mu = 1.0e-5$. Observe in Figures 5.1c and 5.1d that the eigenvalues of matrix $\tilde{N}^{-1}\hat{B}$ are *nicely clustered around one*. On the other hand in Figures 5.1a and 5.1b the eigenvalues of matrix \hat{B} have large variations and there are many small eigenvalues close to zero. Notice that the preconditioner was effective not only at optimality as it was predicted by theory, but through all iterations of pdNCG. This is because starting from the zero solution the iterates W^*x^k were maintained approximately sparse $\forall k$.

Additionally, in Figure 5.2 we show the number of CG/PCG iterations and time required for the unpreconditioned and the preconditioned cases of the same experiment. Observe in Figures 5.2a ($\mu = 1.0e-3$) and 5.2b ($\mu = 1.0e-5$) that the number of PCG iterations are much less than the number of CG iterations. Surprisingly, the number of CG iterations required for the experiment with $\mu = 1.0e-3$ were more than the number of iterations for CG for the experiment with $\mu = 1.0e-5$. Although, matrix \hat{B} has worse condition number in the latter case, see the values of the vertical axis in Figures 5.1a and 5.1b. We believe that this is because of the slightly better clustering of the eigenvalues of matrix \hat{B} for the experiments with $\mu = 1.0e-5$; see Figures 5.1a and 5.1b. Finally, PCG was faster than CG in terms of required time for convergence, see Figures 5.2c and 5.2d.

5.1. Solving systems with the preconditioner. In this subsection we discuss how we can solve systems with the proposed preconditioner \tilde{N} .

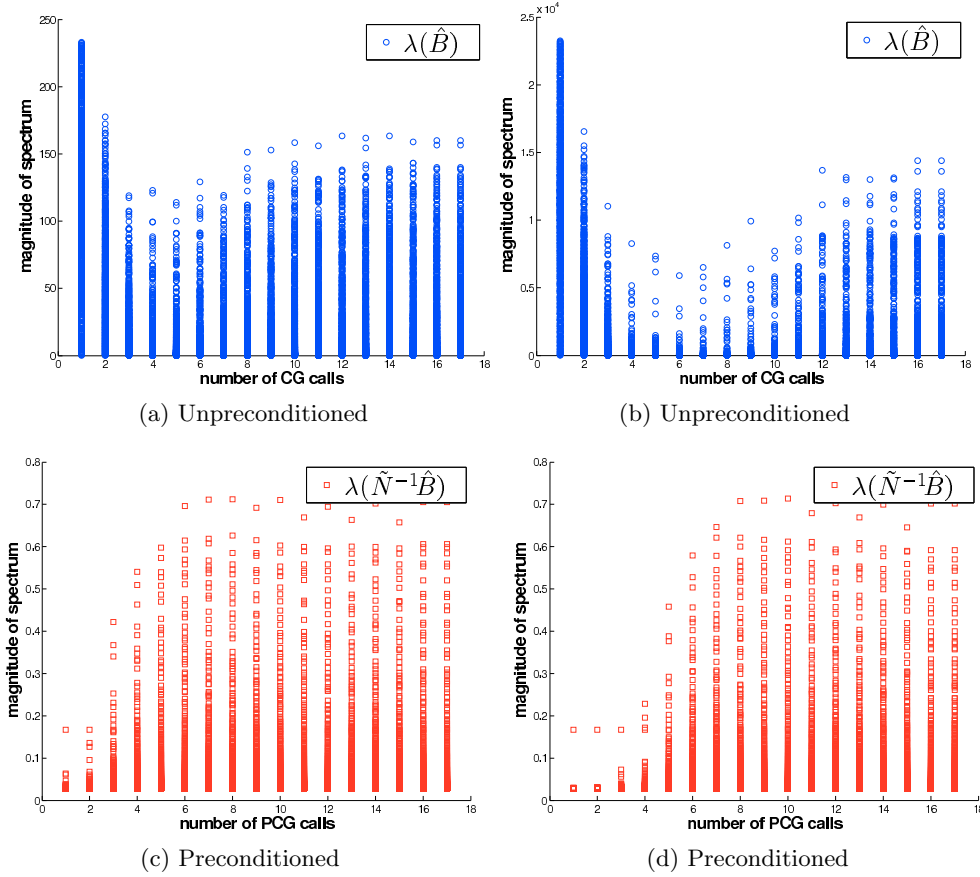


Fig. 5.1: Spectra of $\lambda(\hat{B})$, $\lambda(\tilde{N}^{-1}\hat{B})$ when pdNCG is applied with smoothing parameter $\mu = 1.0e-3$ (left column of sub-figures) and $\mu = 1.0e-5$ (right column of sub-figures). Matrix A in \hat{B} is a 2D DCT, $n = 2^{10}$, $m = n/4$ and $c = 2.29e-2$. Seventeen systems are solved in total for each experiment.

The simplest case is when W is an orthogonal matrix. In this case it is readily verified that solving systems with matrix \tilde{N} costs two matrix vector products with matrices W , W^\top and the inversion of a diagonal matrix. Therefore, this operation is inexpensive. Especially, when matrix W is a DCT or a Wavelets transform for which matrix-vector products with W and W^\top can be calculated in $\mathcal{O}(n \log n)$ and $\mathcal{O}(n)$ time, respectively.

Let us now consider iTV problems. Let p_v be the vertical number of pixels of the image to be reconstructed and p_h be the horizontal number of pixels. For simplicity we will assume that the image is square, hence, $p = p_v = p_h$. Additionally, we assume that the image is handled in a vectorized form, i.e., instead of an image of size $p \times p$ we have a vectorized image of size $p^2 \times 1$ where the columns of the image are stuck one after the other. In this case, for iTV the $W \in \mathbb{C}^{n \times n}$ matrix in problem (1.1) is square with $n = p^2$, rank-deficient with $\text{rank}(W) = n - 1$. Matrix W corresponds to a discretization of the nabla operator and it measures local differences of pixels when

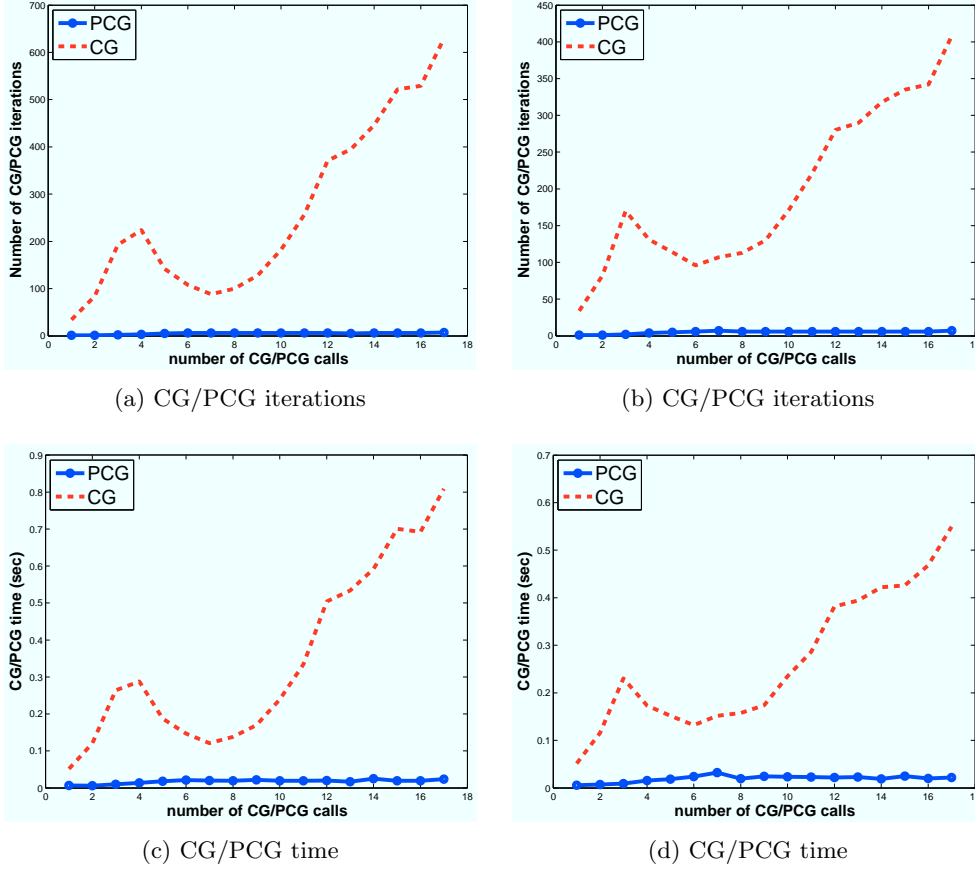


Fig. 5.2: Number of CG/PCG iterations and required time when pdNCG is applied with smoothing parameter $\mu = 1.0e-3$ (left column of sub-figures) and $\mu = 1.0e-5$ (right column of sub-figures). Matrix A in \hat{B} is a 2D DCT, $n = 2^{10}$, $m = n/4$ and $c = 2.29e-2$. Seventeen systems are solved in total for each experiment.

applied on a vectorized image. In particular,

$$W = W_v + \sqrt{-1}W_h,$$

where $W_v \in \mathbb{R}^{n \times n}$ and $W_h \in \mathbb{R}^{n \times n}$. Matrix W_v measures vertical differences of pixels when applied on a vectorized image and it has the following non-zero components:

$$[W_v]_{p(j-1)+i, p(j-1)+i} = -1 \quad \text{and} \quad [W_v]_{p(j-1)+i, p(j-1)+i+1} = 1$$

$\forall j = 1, 2, \dots, p$ and $\forall i = 1, 2, \dots, p-1$. Matrix W_h measures horizontal differences of pixels when applied on a vectorized image and it has the following non-zero pattern:

$$[W_h]_{p(j-1)+i, p(j-1)+i} = -1 \quad \text{and} \quad [W_h]_{p(j-1)+i, p(j-1)+i+p} = 1$$

$\forall j = 1, 2, \dots, p-1$ and $\forall i = 1, 2, \dots, p$.

Observe that matrix \tilde{N} in this case is at most seven-diagonal and it has the following block tridiagonal form

$$(5.14) \quad \tilde{N} = \begin{bmatrix} C_1 & K_1^\top & & & & & \\ K_1 & C_2 & K_2^\top & & & & \\ & K_2 & C_3 & K_3^\top & & & \\ & & K_3 & \ddots & \ddots & & \\ & & & \ddots & C_{p-1} & K_{p-1}^\top & \\ & & & & K_{p-1} & C_p & \end{bmatrix},$$

where $C_i \in \mathbb{R}^{p \times p}$ are tridiagonal matrices $\forall i = 1, 2, \dots, p$ and $K_i \in \mathbb{R}^{p \times p}$ are upper bidiagonal matrices $\forall i = 1, 2, \dots, p-1$. Solving systems with the symmetric positive definite block tridiagonal matrix \tilde{N} can be done in $\mathcal{O}(p^4)$ time by calculating its Cholesky decomposition without re-ordering. More precisely, the Cholesky factor \tilde{L} of $\tilde{N} = \tilde{L}\tilde{L}^\top$ is of the form

$$(5.15) \quad \tilde{L} = \begin{bmatrix} L_1 & & & & & & \\ U_1 & L_2 & & & & & \\ & U_2 & L_3 & & & & \\ & & U_3 & \ddots & & & \\ & & & \ddots & L_{p-1} & & \\ & & & & U_{p-1} & L_p & \end{bmatrix},$$

where $L_i \in \mathbb{R}^{p \times p} \forall i = 1, 2, \dots, p$ and $U_i \in \mathbb{R}^{p \times p} \forall i = 1, 2, \dots, p-1$. The factor \tilde{L} can be calculated by using $\tilde{N} = \tilde{L}\tilde{L}^\top$, (5.14) and (5.15) to get

$$(5.16) \quad L_1 L_1^\top = C_1$$

$$(5.17) \quad U_i L_i^\top = K_i \quad \forall i = 1, 2, \dots, p-1$$

$$(5.18) \quad U_{i-1} U_{i-1}^\top + L_i L_i^\top = C_i \quad \forall i = 2, 3, \dots, p.$$

Notice that C_1 in (5.14) is symmetric positive definite because \tilde{N} is a symmetric positive definite matrix. Since C_1 is symmetric positive definite and tridiagonal we can obtain the lower bidiagonal matrix L_1 in (5.16) by calculating the Cholesky decomposition of matrix C_1 . Calculation of L_1 can be done in $\mathcal{O}(p)$ time. Matrix U_1 is upper diagonal and can be calculated in $\mathcal{O}(p^3)$ time by solving $U_1 L_1^\top = K_1$. The next step is the calculation of L_2 using (5.18), which is the Cholesky factor of $C_2 - U_1 U_1^\top$. The term $C_2 - U_1 U_1^\top$ can be calculated in $\mathcal{O}(p^3)$. Notice that $C_2 - U_1 U_1^\top = C_2 - K_1 C_1^{-1} K_1^\top$ is the Schur complement of a two by two block matrix

$$S = \begin{bmatrix} C_1 & K_1^\top \\ K_1 & C_2 \end{bmatrix}.$$

Matrix S is symmetric positive definite because matrix \tilde{N} is positive definite, this can be readily seen from (5.14). Hence, the Schur complement $C_2 - K_1 C_1^{-1} K_1^\top$ of S is a symmetric positive definite matrix. By repeating this process p times using (5.17) and (5.18) we obtain the matrices L_i and U_i . Since each step requires $\mathcal{O}(p^3)$ time, the total calculation of \tilde{L} requires $\mathcal{O}(p^4)$.

We note that this $\mathcal{O}(p^4)$ operation is expensive for the problems of our interest. Therefore, instead of naively calculating the factor \tilde{L} we employ AMD (Approximate

Table 5.1: Scaling of running time of Cholesky factorization with ADM ordering, p denotes the number of pixels of a $p \times p$ image.

p	16	32	64	128	256	512	1024	2048
CPU (sec)	3.6e-4	1.4e-3	6.1e-3	3.0e-2	1.6e-1	9.1e-1	5.2	30.6

Minimum Degree) ordering [1] by invoking MATLAB’s backslash operator. AMD results in significant reduction in the FLOPS (FLoating-point Operations Per Second) rate and the filling in \tilde{L} . In Table 5.1 we present how the running time of MATLAB’s backslash operator scales as p increases from 2^3 to 2^{11} . The results are averaged over 20 trials. Observe that the running time scales nearly $\mathcal{O}(p^2)$, which is a significant improvement compared to $\mathcal{O}(p^4)$. The matrices \tilde{N} in this experiment were constructed using pdNCG. Additionally, for this experiment we run MATLAB only in one thread in order to eliminate the effects of multithread implementations.

Unfortunately, solving systems with the proposed preconditioner is not always an inexpensive procedure. In particular, solving systems with matrix \tilde{N} when W has orthonormal rows is a non-trivial operation. An example is radar and sonar systems [21], where $W \in \mathbb{R}^{n \times l}$ is a Gabor frame with $n \leq l$. In this case, \tilde{N} does not have a structure which can be exploited in order to solve systems with it inexpensively. In similar cases in the literature, i.e., denoising of images [27], attempts have been made to solve systems with the preconditioner using an iterative method. In our case \tilde{N} is a symmetric positive definite matrix. Therefore, $\text{CG}_{\tilde{N}}$ can be used, where $\text{CG}_{\tilde{N}}$ denotes conjugate gradients method, but we add a subscript \tilde{N} in order to distinguish from the unpreconditioned CG notation for the system in (4.4).

Our personal numerical experience regarding this strategy suggests that it is difficult to control the time required by $\text{CG}_{\tilde{N}}$ to solve systems with \tilde{N} approximately, such that the overall time required by PCG is reduced. Although, the number of PCG iterations is decreased, even with a small number of $\text{CG}_{\tilde{N}}$ iterations. In Figure 5.3 we present an example where using the preconditioner \tilde{N} in an iterative fashion does not decrease the time required by pdNCG. The tested problem is a radar tone reconstruction instance [5], where $W \in \mathbb{R}^{n \times l}$ is a Gabor frame with $n = 2^{13}$, $l = 228864$, $\mu = 1.0e-5$, $c = 2.41e-5$ and $\rho = 5.0e-1$. Matrix A is a block-diagonal with ± 1 entries in the blocks and $m = 648$ rows. We make four experiments. For the first three experiments we vary the number of $\text{CG}_{\tilde{N}}$ iterations for solving systems with \tilde{N} . The number of $\text{CG}_{\tilde{N}}$ iterations is set to 5, 10 and 20, respectively for each experiment. The last experiment is using unpreconditioned CG. Observe in Figure 5.3a that PCG requires significantly fewer iterations than unpreconditioned CG for all settings of $\text{CG}_{\tilde{N}}$. However, notice in Figure 5.3b the time required by PCG is larger than the time required by CG.

6. Continuation. In the previous section we have shown that by using preconditioning, the spectral properties of systems which arise can be improved. However, for initial stages of pdNCG a similar result can be achieved without the cost of having to apply preconditioning. In particular, at initial stages the spectrum of \hat{B} can be controlled to some extent through inexpensive continuation. Whilst preconditioning is enabled only at later stages of the process. Briefly by continuation it is meant that a sequence of “easier” subproblems is solved, instead of solving directly problem (2.2). The reader is referred to Chapter 11 in [25] for a survey on continuation methods in

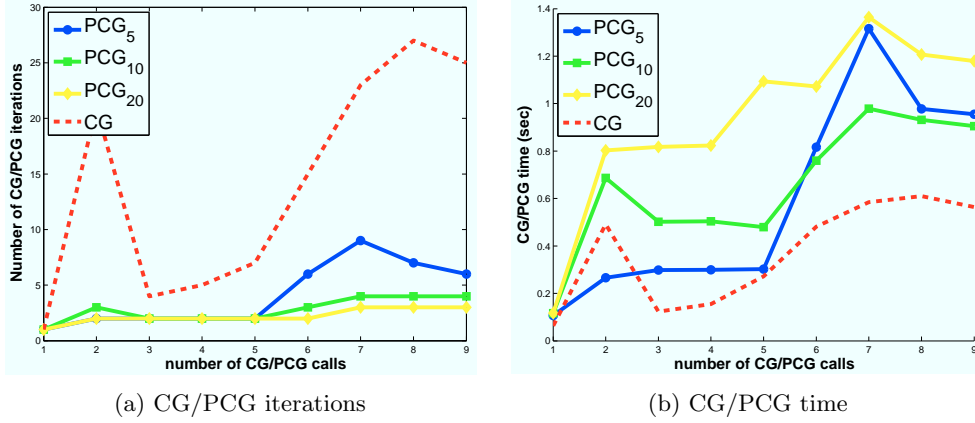


Fig. 5.3: Number of CG/PCG iterations and required time when $\mu = 1.0e-5$ and systems with the preconditioner \tilde{N} are solved approximately using conjugate gradients. PCG₅, PCG₁₀ and PCG₂₀ correspond to PCG, where systems are solved with \tilde{N} approximately using conjugate gradients which is terminated after 5, 10 and 20 iterations, respectively.

- 1: **Outer loop:** For $j = 0, 1, 2, \dots, \vartheta$, produce $(c^j, \mu^j)_{j=0}^{\vartheta}$.
- 2: **Inner loop:** Approximately solve the subproblem

$$\text{minimize } f_{c^j}^{\mu^j}(x)$$

using pdNCG and by initializing it with the solution of the previous subproblem.

Fig. 6.1: Continuation framework

optimization.

In this paper we use a similar continuation framework to [4, 10, 11, 17]. In particular, a sequence of sub-problems (2.2) is solved, where each of them is parameterized by c and μ simultaneously. Let \tilde{c} and $\tilde{\mu}$ be the final parameters for which problem (2.2) must be solved. Then the number of continuation iterations ϑ is set to be the maximum order of magnitude between $1/\tilde{c}$ and $1/\tilde{\mu}$. For instance, if $\tilde{c} = 1.0e-2$ and $\tilde{\mu} = 1.0e-5$ then $\vartheta := \max(2, 5) = 5$. If $\vartheta \geq 2$, then the initial parameters c^0 and μ^0 are both always set to $1.0e-1$ and the intervals $[c^0, \tilde{c}]$ and $[\mu^0, \tilde{\mu}]$ are divided in ϑ equal subintervals in logarithmic scale. For all experiments that we have reported in this paper we have found that this setting leads to a generally acceptable improvement over pdNCG without continuation. The pseudo-code of the proposed continuation framework is shown in Figure 6.1.

Figure 6.2 shows the performance of pdNCG for four cases, no continuation and no preconditioning, no continuation with preconditioning, continuation with preconditioning through the whole process and continuation with preconditioning only at later stages. The vertical axis of Figure 6.2 shows the relative error $\|x^k - x_{\tilde{c}, \tilde{\mu}}\|_2 / \|x_{\tilde{c}, \tilde{\mu}}\|_2$.

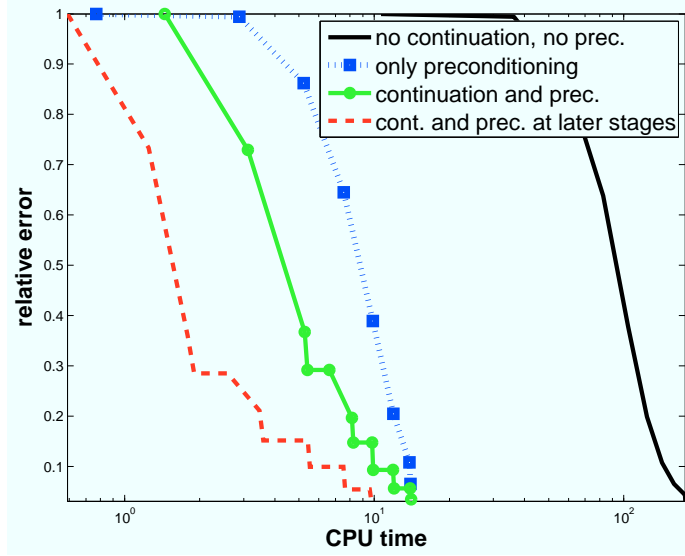


Fig. 6.2: Performance of pdNCG for four different settings, i) no continuation and no preconditioning, ii) no continuation with preconditioning, iii) continuation with preconditioning through all iterations and iv) continuation with preconditioning only at later stages. The vertical axis presents the relative error $\|x^k - x_{\tilde{c}, \tilde{\mu}}\|_2 / \|x_{\tilde{c}, \tilde{\mu}}\|_2$, where $x_{\tilde{c}, \tilde{\mu}}$ is the optimal solution for the parameter setting \tilde{c} , $\tilde{\mu}$ in problem (2.2). The horizontal axis is in log-scale.

The optimal $x_{\tilde{c}, \tilde{\mu}}$ is obtained by using pdNCG with parameter tuning set to recover a highly accurate solution. The horizontal axis shows the CPU time. The problem is an iTV problem where matrix A is a partial $2D$ DCT, $n = 2^{16}$, $m = n/4$, $c = 5.39e-2$ and $\rho = 5.0e-1$. The final smoothing parameter $\tilde{\mu}$ is set to $1.0e-5$. For the experiment that preconditioning is used only at later stages of continuation; preconditioning is enabled when $\mu^j \leq 1.0e-4$, where j is the counter for continuation iterations. All experiments are terminated when the relative error $\|x^k - x_{\tilde{c}, \tilde{\mu}}\|_2 / \|x_{\tilde{c}, \tilde{\mu}}\|_2 \leq 1.0e-1$. Solving approximately the problem is an acceptable practice since the problem is very noisy (i.e. signal-to-noise-ratio is 10 decibel) and there is not much improvement of the reconstructed image if more accurate solutions are requested. Finally, all other parameters of pdNCG were set to the same values for all four experiments. Observe in Figure 6.2 that continuation with preconditioning only at late stages was the best approach for this problem.

7. Numerical experiments. In this section we demonstrate the efficiency of pdNCG against state-of-the-art methods for CS. We briefly discuss existing methods, describe the setting of the experiments and finally present numerical results. All experiments that are demonstrated in this paper can be reproduced by downloading the software from <http://www.maths.ed.ac.uk/ERGO/pdNCG/>.

7.1. Existing algorithms. We compare pdNCG with three state-of-the-art first-order methods, TFOCS [5], TVAL3 [19] and TwIST [6].

- TFOCS (Templates for First-Order Conic Solvers) is a MATLAB software for the solution of signal reconstruction problems. TFOCS solves the dual

problem of

$$(7.1) \quad \text{minimize } c\|W^*x\|_1 + \frac{\mu_{T_1}}{2}\|x - x^0\|_2^2 + \frac{1}{2}\|Ax - b\|_2^2,$$

where μ_{T_1} is a positive constant. TFOCS also solves the dual problem of

$$(7.2) \quad \begin{aligned} & \min_{x \in \mathbb{R}^n} \quad \|W^*x\|_1 + \frac{\mu_{T_2}}{2}\|x - x^0\|_2^2 \\ & \text{subject to: } \|Ax - b\|_2 \leq \epsilon, \end{aligned}$$

where $\epsilon > 0$. Although problems (7.1) and (7.2) are non-smooth, the regularization terms $\mu_{T_1}/2\|x - x^0\|_2^2$ and $\mu_{T_2}/2\|x - x^0\|_2^2$ yield smooth convex dual problems, which can be solved by standard first-order methods. In particular, the smooth dual problems are solved using the Auslender and Teboulle's accelerated first-order method [2]. In our experiments we present results for TFOCS for both problems (7.1) and (7.2). We denote by TFOCS_unc the version that solves the unconstrained problem (7.1) and by TFOCS_con the version that solves the constrained problem (7.2). TFOCS can be downloaded from [5].

- TVAL3 (Total-Variation minimization by Augmented Lagrangian and ALternating direction ALgorithms) is a MATLAB software for the solution of signal reconstruction problems regularized with the total-variation semi-norm. TVAL3 reformulates problem (1.1) to the equivalent problem

$$(7.3) \quad \text{minimize } \sum_{i=1}^l \|\Omega_i^T x\|_2 + \frac{1}{2c}\|Ax - b\|_2^2,$$

where $\Omega_i = [ReW_i, ImW_i] \in \mathbb{R}^{n \times 2}$. Then it solves the augmented Lagrangian reformulation of problem (7.3), which is

$$(7.4) \quad \text{minimize } \sum_{i=1}^l (\|u_i\|_2 + \frac{\beta}{2}\|\Omega_i^T x - u_i\|_2^2 - v_i^T(\Omega_i^T x - u_i)) + \frac{1}{2c}\|Ax - b\|_2^2,$$

where $u_i, v_i \in \mathbb{R}^2$ and β, c are positive constants. The augmented Lagrangian in (7.4) is minimized for variables $x \in \mathbb{R}^n$ and u_i $i = 1, 2, \dots, l$. The parameters v_i $i = 1, 2, \dots, l$ are handled by the method.

- TwIST (Two-step Iterative Soft Thresholding): is also a MATLAB software for signal/image processing problems. TwIST solves problem (1.1). TwIST is a nonlinear two-step iterative version of IST, which according to its authors is more effective on ill-conditioned and ill-posed problems.

Another solver is NestA [4] (by the same authors as TFOCS) which can also solve (1.1) but it is applicable only in the case that $(AA^T)^{-1}$ is available. Additionally, TFOCS is the successor of NestA, since both apply the similar techniques but TFOCS is a newer and allows of more control options. Another method is the Primal-Dual Hybrid Gradient (PDHG) of [14]. PDHG has been reported to be very efficient for imaging applications such as denoising and deblurring, for which matrix A is the identity or a square and full-rank matrix which is inexpensively diagonalizable. Unfortunately, this is not the case for the CS problems that we are interested in. However, for all earlier mentioned methods the matrix inversion can be replaced with a solution of a linear system at every iteration of the methods or a one-time cost of a

factorization. To the best of our knowledge, there are no available implementations with such modifications for these methods.

There exists also a generic proximal algorithm for total-variation [12] and the Generalized Iterative Soft Thresholding (GISTA) in [20] for which we do not have generic implementations for CS problems.

7.2. Equivalent problems. Solvers pdNCG, TFOCS_unc, TVAL3 and TwIST solve the penalized least squares problem (1.1), while TFOCS_con solves the constrained least squares problem (7.2).

In our experiments we put significant effort in calibrating the parameters c and ϵ for the penalized and constrained least-squares problems, respectively, such that all methods solve similar problems. First, we set $\epsilon = \|b - \tilde{b}\|_2$ in (7.2), where \tilde{b} is the noiseless sampled signal. Hence problem (7.2) is parameterized with the optimal ϵ . Then we find an approximation of the optimal c . By optimal c we mean the value of c for which problems (1.1) and (7.2) are equivalent if $\epsilon = \|b - \tilde{b}\|_2$ and $\mu_{T_2} = 0$. Let ω denote the optimal Lagrange multiplier of (7.2). If $\epsilon = \|b - \tilde{b}\|_2$ and $\mu_{T_2} = 0$, then it is easy to show that for $c := 2/\omega$ problems (1.1) and (7.2) are equivalent.

The exact optimal Lagrange multiplier ω is not known a-priori. However it can be calculated by solving to high accuracy the dual problem of (7.2) with TFOCS by setting $\mu_{T_2} \approx 0$ in (7.2). Then we set $c := 2/\omega$. If \tilde{b} is not available, then ϵ is set such that a visually pleasant solution is obtained.

7.3. Parameter tuning and hardware. The parameter μ of pdNCG is set to $\mu = 1.00e-5$, which for the problems of our interest resulted in solutions with the similar or better accuracy than the compared methods. The parameter η in (4.6) is set to $1.0e-1$, the maximum number of backtracking line-search iterations is fixed to 10. Moreover, the backtracking line-search parameters τ_1 and τ_2 in step 5 of pdNCG (Fig. 4.1) are set to $9.0e-1$ and $1.0e-3$, respectively. The constant ρ of the preconditioner in (5.2) is set to $5.0e-1$.

For TVAL3 parameter β is set to $\beta = 2^8$ based on suggestions of its authors in [19] and our personal experience. Moreover, continuation was enabled in order to enhance the performance of the method. Any other parameters that were not discussed are set to their default values.

We tune TwIST based on comments/suggestions of its authors and personal experience. In particular, parameter λ is set to $\lambda = 0.04$ and the maximum number of iterations for the iTV denoising procedure is set to 10.

The version 1.3.1 of TFOCS has been used. The termination criterion of TFOCS is by default the relative step-length. The tolerance for this criterion is set to the default value, except in cases that certain suggestions are made in TFOCS software package or the corresponding paper [5]. The default Auslender and Teboulle's single-projection method is used as a solver for TFOCS. Moreover, as suggested by the authors of TFOCS, appropriate scaling is performed on matrices A and W , such that they have approximately the same Euclidean norms. All other parameters are set to their default values, except in cases that specific suggestions are made by the authors. Generally, regarding tuning of TFOCS, substantial effort has been made to guarantee that problems are not over-solved.

All solvers are MATLAB implementations and all experiments are performed on a MacBook Air running OS X 10.10.1 with 2 GHz (3 GHz turbo boost) Intel Core Duo i7 processor using MATLAB R2012a. The cores were working with frequency 2.7 - 3 GHz during the experiments and we did not observe any CPU throttling.

7.4. Termination criteria. For images we measure the quality of the reconstructed solutions by using the Peak-Signal-to-Noise-Ratio (PSNR) function

$$\text{PSNR} = 10 \log_{10} \left(\frac{\text{peakval}^2}{\text{MSE}} \right),$$

where peakval is the range of the image datatype, in this case the range is one since we work with black and white images, and MSE is the mean squared error between the solution and the original noiseless image. For other types of signals we measure the quality of the reconstructed solutions by measuring their Signal-to-Noise-Ratio (SNR).

We terminate pdNCG, TVAL3 and TwIST when the PSNR (for images) or the SNR (for other types of signals) of their solution is equal or larger than the PSNR or SNR of the solution obtained by TFOCS_unc. This way we make sure that all methods which solve the penalized least-squares problem terminate when a solution of the same quality as the one of TFOCS_unc is obtained. As we mentioned in Subsection 7.3 when we use TFOCS_unc we do not over-solve the problem, otherwise, we would favour pdNCG, which is a second-order method. Since pdNCG, TVAL3, TwIST and TFOCS_unc solve the same problem with the same penalty parameter c then we can make a fair comparison of their performance. The solution obtained by TFOCS_con might differ slightly in terms of PSNR or SNR compared to the ones obtained by pdNCG, TVAL3, TwIST and TFOCS_unc. However, this is because we set parameter c to be approximately close to the optimal value that makes the penalized and the constrained problems equivalent.

7.5. Problems sets. We compare the solvers pdNCG, TFOCS, TVAL3 and TwIST on image reconstruction problems which are modelled using iTV. We separate the images to be reconstructed into two sets, which are shown in Figures 7.1 and 7.2. Figure 7.1 includes some standard images from the image processing community. There are seven images in total, the house and the peppers, which have 256×256 pixels and Lena, the fingerprint, the boat and Barbara, which have 512×512 pixels. Finally, the image Shepp-Logan has variable size depending on the experiment. Figure 7.2 includes images which have been sampled using a single-pixel camera [13]. Briefly a single-pixel camera samples random linear projections of pixels of an image, instead of directly sampling pixels. The problem set can be downloaded from <http://dsp.rice.edu/cscamera>. In this set there are in total five sampled images, the dice, the ball, the mug, the letter R and the logo. Each image has 64×64 pixels.

Moreover, we present the performance of pdNCG and TFOCS on the recovery of radio-frequency radar tones. This problem has been first demonstrated in Subsection 6.5 of [5]. We describe again the setting of the experiment. The signal to be reconstructed consists of two radio-frequency radar tones which overlap in time. The amplitude of the tones differs by 60 dB. The carrier frequencies and phases are chosen uniformly at random. Moreover, noise is added such that the larger tone has SNR 60 dB and the smaller tone has SNR 2.1e-2 dB. The signal is sampled at 2^{15} points, which corresponds to Nyquist sampling rate for bandwidth 2.5 GHz and time period approximately 6.5e+3 ns. The reconstruction is modelled as a CS problem where matrix $A \in \mathbb{R}^{m \times n}$ is block-diagonal with ± 1 for entries, $n = 2^{15}$ and $m = 2616$, i.e. subsampling ratio $m/n \approx 7.9e-3$. Moreover, $W \in \mathbb{R}^{n \times l}$ is a Gabor frame with $l = 915456$.

7.6. Dependence of pdNCG on smoothing parameter. In this subsection we present the performance of pdNCG with and without preconditioning for decreas-

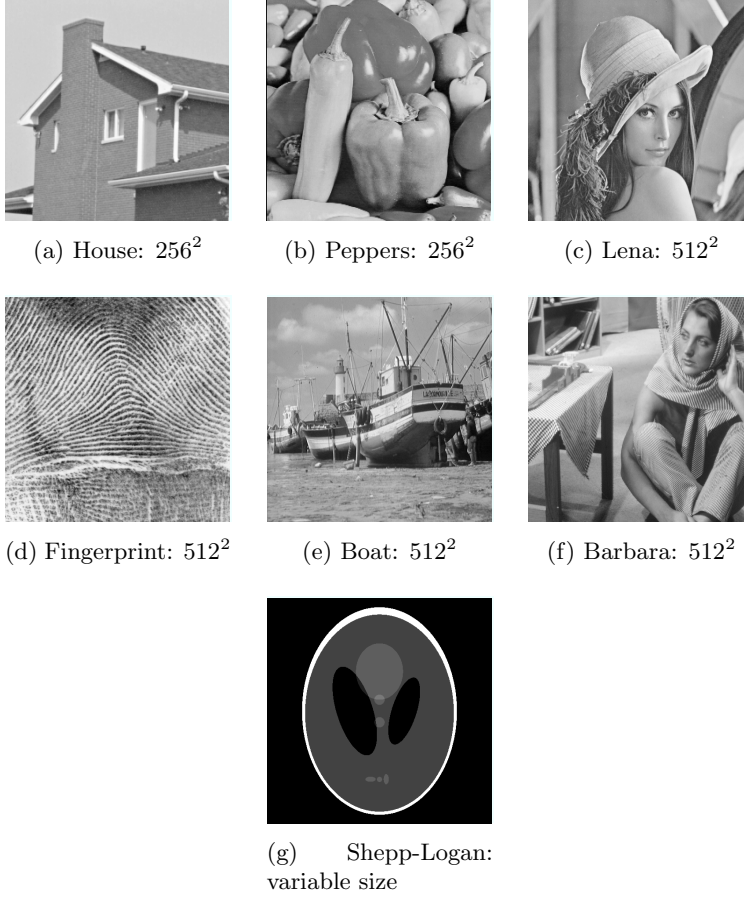


Fig. 7.1: Benchmark images, the number of pixels for each image is given in the sub-captions . For Figure 7.1g the size varies depending on the experiment

ing values of the smoothing parameter μ . For this experiment we use the images from Figures 7.1a to 7.1f. The CS matrix for all experiments is a partial Discrete Cosine Transform (DCT) matrix with $m \approx n/4$ and n is equal to the number of pixels of each image in Figure 7.1. For all experiments the sampled signals have PSNR equal to 20 decibels (dB).

The results of the experiments are shown in Table 7.1. In Table 7.1 notice that for the preconditioned case there is always a large increase in CPU time from $\mu = 1.0e-02$ to $1.0e-04$. This is because for $\mu = 1.0e-02$ pdNCG relies only on continuation, while for values of μ equal or smaller than $1.0e-04$ preconditioning is necessary and it is automatically activated using the technique described in Section 6. Overall pdNCG had a stable performance with respect to the smoothing parameter μ . This is due to the good performance of the proposed preconditioner. Notice that without the preconditioner the performance of pdNCG for $\mu \leq 1.0e-04$ worsens noticeably. In particular, for some experiments the unpreconditioned pdNCG required more than 3 hours of CPU time. For these experiments we forced termination of the method and

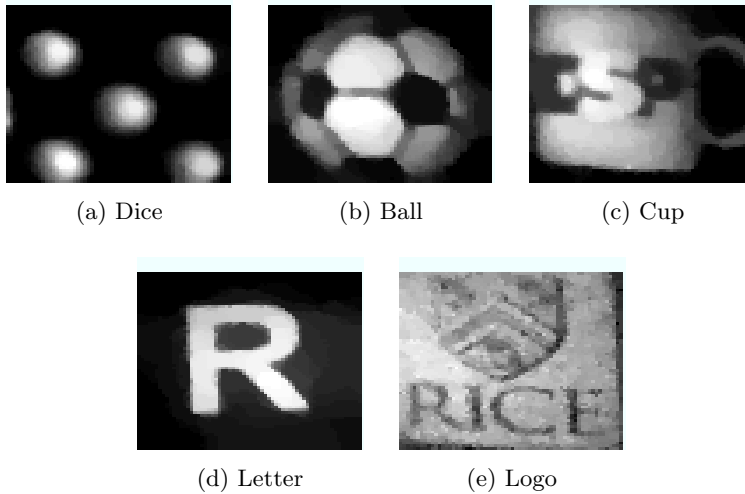


Fig. 7.2: Benchmark images, which were sampled using the single-pixel camera [13]

we do not report any results.

7.7. Dependence on problem size. We now present the performance of methods pdNCG, TFOCS, TVAL3 and TwIST as the size of the problem n increases. The image from Figure 7.1g has been used for this experiment. Again, the CS matrix for all experiments is a partial Discrete Cosine Transform (DCT) matrix with $m \approx n/4$. The sampled signals have PSNR equal to 20 dB.

The results of this experiment are shown in Table 7.2. Observe that all methods exhibit a linear-like increase in CPU time as a function of the size of the problem. We denote with bold the problem for which pdNCG was the fastest method.

In this table we present the PSNR of the recovered solutions for each solver. Notice that TVAL3 does not converge always to a solution of similar PSNR as the other solvers. Although, we put significant effort to tune its parameters. A similar performance of TVAL3 is observed for many of the experiments in the subsequent subsections. Moreover, observe that TwIST was much slower than the other methods and it did not converge to a solution of similar PSNR to TFOCS or pdNCG for all experiments. Similar performance for TwIST has been observed in [19].

7.8. Dependence on the level of noise. In this experiment we compare the solvers pdNCG, TFOCS and TVAL3 as the level of noise increases. We exclude TwIST from this and subsequent experiments due to its poor performance on the simple synthetic experiment reported in Subsection 7.7; similar performance has been observed in [19]. For this experiment we use the images from Figures 7.1a to 7.1f. The CS matrix for all experiments is a partial Discrete Cosine Transform (DCT) matrix with $m \approx n/4$.

In Table 7.3 we present the performance of the methods for this experiments. In the second column of Table 7.3 the PSNR is shown, which is decreasing from 95 dB to 20 dB in six steps. The rest of the table shows the CPU time, which was required by each solver. Overall pdNCG has good performance for problems with large level of noise, i.e., PSNR equal to 20 dB. We denote with bold the problems

Table 7.1: Performance of pdNCG for decreasing values of the smoothing parameter μ . For this experiment the images from Figures 7.1a to 7.1f have been used. The table shows the CPU time in seconds required for preconditioned pdNCG and unpreconditioned pdNCG for each combination of μ and problem. PSNR corresponds to the reconstructed solution of pdNCG.

μ	CG/PCG	House	Peppers	Lena	Fingerprint	Boat	Barbara
1.0e-02	PCG	4	4	22	19	23	19
	CG	4	4	22	19	22	19
	PSNR	19.2	24.4	25.6	18.1	24	22.6
1.0e-04	PCG	8	9	40	136	50	42
	CG	10	11	85	155	85	57
	PSNR	19.2	24.4	25.6	18.1	24	22.6
1.0e-07	PCG	10	10	56	203	57	85
	CG	82	98	1165	2839	1519	852
	PSNR	19.3	24.4	25.6	18.1	24	22.6
1.0e-10	PCG	11	12	73	182	76	86
	CG	273	195	3484	-	3208	3508
	PSNR	19.3	24.4	25.6	18.1	24	22.6
1.0e-13	PCG	11	12	66	232	68	85
	CG	265	242	4834	-	5356	-
	PSNR	19.3	24.4	25.6	18.1	24	22.6

Table 7.2: Performance of pdNCG, TFOCS, TVAL3 and TwIST for increasing problem size. The image Shepp-Logan from Figure 7.1g has been used for this experiment. The table shows the required CPU time and the PSNR of the recovered solutions for each solver.

Solver	n=	64 ²	128 ²	256 ²	512 ²	1024 ²
TFOCS _{con}	CPU time (sec)	16	23	55	264	1034
	PSNR	15.8	17.4	17.9	18	18
TFOCS _{unc}	CPU time (sec)	19	30	79	385	1477
	PSNR	15.8	17.4	17.9	18	18
TVAL3	CPU time (sec)	1	2	250	1365	4843
	PSNR	15.8	17.4	17.2	17.2	17.3
TwIST	CPU time (sec)	28	135	259	2149	7223
	PSNR	13.5	15.9	16.8	16.9	16.9
pdNCG	CPU time (sec)	2	6	13	62	237
	PSNR	15.8	17.4	17.9	18	18

for which pdNCG was the fastest solver. In this table we use the star superscript to denote solvers, which solve the unconstrained problem (1.1) but do not converge to a solution of equal or larger PSNR than the solutions of TFOCS_{unc}.

In Table 7.4 we show the PSNR for the solutions calculated by TFOCS_{con} for the corresponding experiments in Table 7.3. For experiments that are not denoted

Table 7.3: Performance of pdNCG, TFOCS and TVAL3 for increasing level of noise (decreasing PSNR). PSNR is measured in dB. For this experiment the images from Figures 7.1a to 7.1f have been used. The table shows the CPU time in seconds required by each solver. By the star superscript we denote methods that failed to converge to a similar solution as the other solvers.

Solver	PSNR	House	Peppers	Lena	Fingerprint	Boat	Barbara
TFOCS _{con}	95	55	55	262	261	263	263
	80	55	55	261	263	265	262
	65	55	55	262	264	265	262
	50	55	55	262	262	262	262
	35	55	55	261	262	262	262
	20	54	55	263	262	262	261
TFOCS _{unc}	95	76	76	382	381	390	382
	80	76	76	383	385	388	381
	65	76	76	384	399	386	381
	50	75	76	383	383	382	381
	35	76	76	382	382	383	382
	20	76	76	382	379	380	379
TVAL3	95	3	3	16	10	38	55
	80	3	3	16	10	38	55
	65	3	4	10	10	37	56
	50	3	3	383	10	24	44
	35	3	3	1104*	8	36	1067*
	20	184*	185*	999*	8	10	987*
pdNCG	95	20	82	145	141	182	209
	80	20	82	146	142	182	208
	65	20	243	145	142	178	209
	50	20	55	169	142	116	209
	35	13	38	109	138	111	202
	20	16	16	101	156	103	105

with a star superscript in Table 7.3 the solvers pdNCG, TFOCS_{unc} and TVAL3 obtained solutions of similar PSNR due to our setting described in Subsection 7.2.

7.9. Dependence on number of measurements. In this experiment we compare the three methods for a decreasing number of measurements m . For this experiment we use the images from Figures 7.1a to 7.1f. The CS matrix is a partial Discrete Cosine Transform (DCT) matrix. For all experiments the sampled signals have PSNR equal to 20 dB.

The results of this experiment are shown in Table 7.5. We denote with bold the problems for which pdNCG was the fastest method. In Table 7.6 we show the PSNR for the solutions calculated by TFOCS_{con} for the corresponding experiments in Table 7.5. For experiments that are not denoted with a star superscript in Table 7.5 the solvers pdNCG, TFOCS_{unc} and TVAL3 obtained solutions of similar PSNR due to our setting described in Subsection 7.2.

Table 7.4: This table shows the PSNR for the solutions calculated by TFOCS_con for the corresponding experiments in Table 7.3 For this experiment the images from Figures 7.1a to 7.1f have been used.

Solver	PSNR	House	Peppers	Lena	Fingerprint	Boat	Barbara
TFOCS_con	95	20	31.9	29.5	20.1	27.6	25
	80	20	31.9	29.5	20.1	27.6	25
	65	20	31.8	29.5	20.1	27.6	25
	50	20	31.6	29.5	20.1	27.5	24.9
	35	19.9	29.7	28.6	19.7	26.9	24.7
	20	19.2	24.3	25.6	17.9	24	22.6

Table 7.5: Performance of pdNCG, TFOCS and TVAL3 for decreasing number of measurements m . In the second column the percentage of measurements is shown, for example 75% means that $m \approx 3n/4$, where n is the number of pixels in the image to be reconstructed. For this experiment the images from Figures 7.1a to 7.1f have been used. The table shows the CPU time in seconds required by each solver. By the star superscript we denote methods that failed to converge to a similar solution as the other solvers.

Solver	m	House	Peppers	Lena	Fingerprint	Boat	Barbara
TFOCS_con	75%	58	58	273	272	272	273
	50%	56	56	268	266	268	270
	25%	54	55	263	261	265	263
TFOCS_unc	75%	78	78	392	388	393	396
	50%	77	77	389	383	387	389
	25%	76	76	382	378	386	380
TVAL3	75%	296*	301*	1584*	1587*	1497*	1250*
	50%	262*	253*	8	7	7	10
	25%	184*	184*	998*	8	10	986*
pdNCG	75%	23	23	132	134	136	140
	50%	21	21	131	162	135	137
	25%	16	16	101	156	104	105

7.10. Single-pixel camera. We now compare TFOCS with pdNCG on realistic image reconstruction problems where the data have been sampled using a single-pixel camera [13]. In this experiment we compare our solver only with TFOCS_con. This is because in all previous experiments TFOCS_con was faster than TFOCS_unc. Additionally, we were not able to make TVAL3 to converge to a solution which was as visually pleasant as the solutions obtained by TFOCS_con and pdNCG. We believe that this is due to the different CS matrix A in these experiments. In particular, matrix $A \in \mathbb{R}^{m \times n}$, where $n = 64^2$ and $m \approx 0.4n$, is a partial Walsh basis which takes values 0/1 instead of ± 1 . We noticed that this matrix A does not satisfy the RIP property in Definition 1.1 with small δ_q . Therefore, the least squares term in problem (1.1) might be ill-conditioned and this causes difficulties for TVAL3.

Moreover the optimal solutions are unknown and additionally the level of noise is

Table 7.6: This table shows the PSNR for the solutions calculated by TFOCS_con for the corresponding experiments in Table 7.5. For this experiment the images from Figures 7.1a to 7.1f have been used.

Solver	m	House	Peppers	Lena	Fingerprint	Boat	Barbara
TFOCS_con	75%	19.6	27.9	27.2	19.9	25.9	24.8
	50%	19.5	26.6	26.7	19.2	25.2	23.8
	25%	19.2	24.3	25.6	17.9	24	22.6

unknown. Hence the reconstructed images can only be compared by visual inspection. For all four experiments 40% of measurements are selected uniformly at random.

The reconstructed images by the solvers TFOCS_con and pdNCG are presented in Figure 7.3. Solver pdNCG was faster on four out of five problems. On problems that pdNCG was faster it required on average 1.5 times less CPU time. Although it would be possible to tune pdNCG such that it is faster on all problems, we preferred to use its (simple) default tuning in order to avoid a biased comparison.

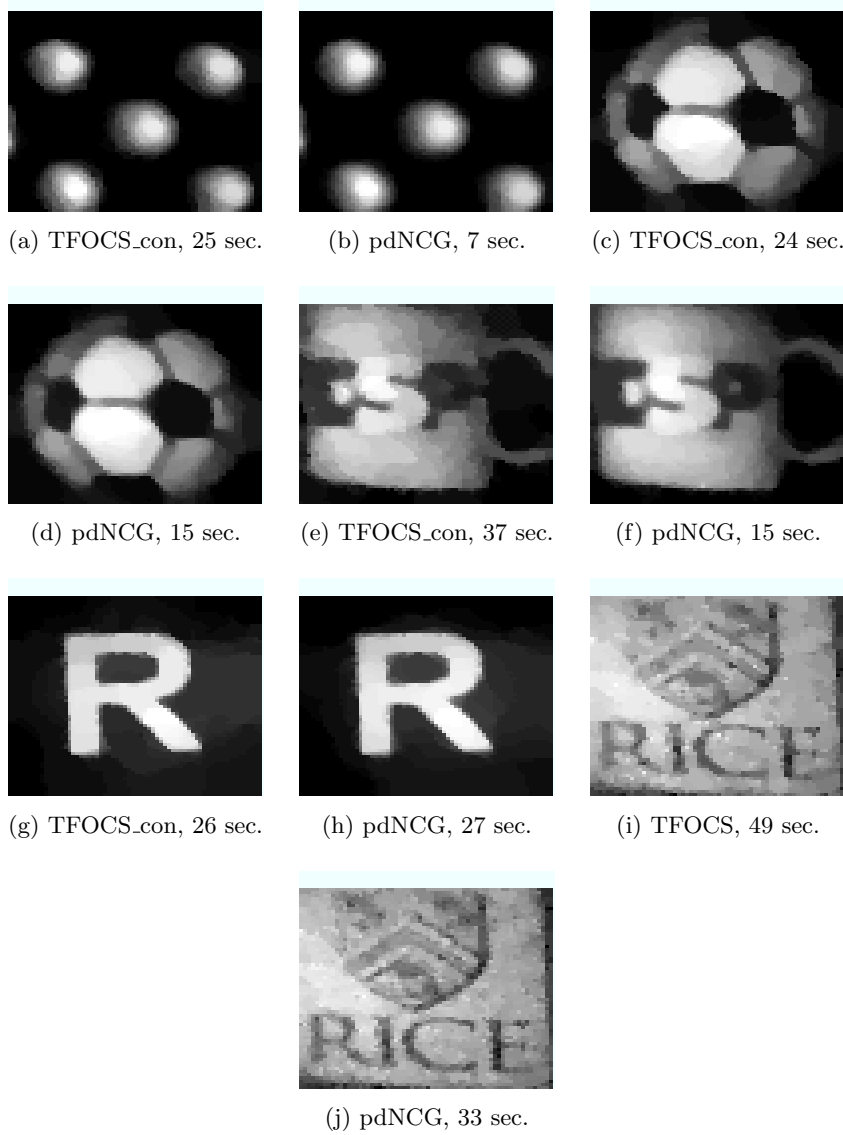


Fig. 7.3: Experiment on realistic image reconstruction where the samples are acquired using a single-pixel camera. The subcaptions of the figures show the required seconds of CPU time for the image to be reconstructed for each solver.

7.11. Radar tone reconstruction. In this subsection we present the performance of TFOCS_unc and pdNCG for the radar tone reconstruction problem, which was described in Subsection 7.5. We exclude TVAL3 from this experiment because it is implemented to solve only TV problems. We also exclude TFOCS_unc since it is superseded in all previous experiments by TFOCS_con.

The results of the comparison are presented in Figure 7.4. Observe in Figures

7.4a and 7.4b that both solvers recovered a solution of similar accuracy but pdNCG was slightly faster. The solution of TFOCS_con has SNR 62.3 dB and the solution of pdNCG has SNR 64.7 dB. It is important to mention that the problems were not over-solved. TFOCS_con was tuned as suggested by its authors in a similar experiment which is shown in Subsection 6.5 of [5].

In Figure 7.4c we plot the SNR against CPU time for every iteration of pdNCG and TFOCS_con. Observe that nearly for all iterations pdNCG the approximate solutions of pdNCG had larger SNR than the approximate solutions of TFOCS_con.

For this experiment we enabled preconditioning for pdNCG. Although, in Subsection 5.1 we mentioned that preconditioning might affect adversely the performance of pdNCG in terms of CPU time. However, we noticed that by enabling preconditioning pdNCG was more stable. Therefore, we believe that it is worth paying the cost of a slight increase of a CPU time to improve the overall robustness of pdNCG.

8. Conclusions. Recently there has been great interest in the development of optimization methods for the solution of compressed sensing problems. The methods that have been developed so far are mainly first-order methods. This is because first-order methods have inexpensive iterations and frequently offer fast initial progress in the optimization process. On the contrary, second-order methods are considered to be rather expensive. The reason is that often access to second-order information requires the solution of linear systems. In this paper we develop a second-order method, a primal-dual Newton Preconditioned Conjugate Gradients. We show that an *approximate* solution of linear systems which arise is sufficient to speed up an iterative method and additionally make it more robust. Moreover, we show that for compressed sensing problems an inexpensive preconditioner can be designed that speeds up even further the approximate solution of linear systems. Extensive numerical experiments are presented which support our findings. Spectral analysis of the preconditioner is performed and shows its very good limiting behaviour.

Acknowledgement. We are grateful to the anonymous reviewers who made crucial recommendations which improved the clarity of contribution and the quality of this paper.

REFERENCES

- [1] P. R. Amestoy. Algorithm 837: Amd, an approximate minimum degree ordering algorithm. *ACM Transactions on Mathematical Software (TOMS)*, 30(3):381–388, 2004.
- [2] A. Auslender and Teboulle. Interior gradient and proximal methods for convex and conic optimization. *SIAM Journal on Optimization*, 16(3):697–725, 2006.
- [3] A. Beck and M. Teboulle. Smoothing and first order methods: A unified framework. *SIAM J. Optim.*, 22(2):557–580, 2012.
- [4] S. R. Becker, J. Bobin, and E. J. Candès. NestA: A fast and accurate first-order method for sparse recovery. *SIAM J. Imaging Sciences*, 4(1):1–39, 2011.
- [5] S. R. Becker, E. J. Candès, and M. C. Grant. Templates for convex cone problems with applications to sparse signal recovery. *Mathematical Programming Computation*, 3(3):165–218, 2011. <http://cvxr.com/tfocs/>.
- [6] J. M. Bioucas-Dias and M. A. T. Figueiredo. A new TwIST: Two-step iterative shrinkage/thresholding algorithms for image restoration. *IEEE Transactions on Image Processing*, 16(12):2992–3004, 2007.
- [7] E. J. Candès. The restricted isometry property and its implications for compressed sensing. *C. R. Acad. Sci. Paris*, 346:589–592, 2008.
- [8] E. J. Candès and D. L. Donoho. New tight frames of curvelets and optimal representations of objects with piecewise c^2 singularities. *Comm. Pure Appl. Math.*, 57:219–266, 2004.
- [9] E. J. Candès, Y. C. Eldar, and D. Needell. Compressed sensing with coherent and redundant dictionaries. *Applied and Computational Harmonic Analysis*, 31(1):59–73, 2011.

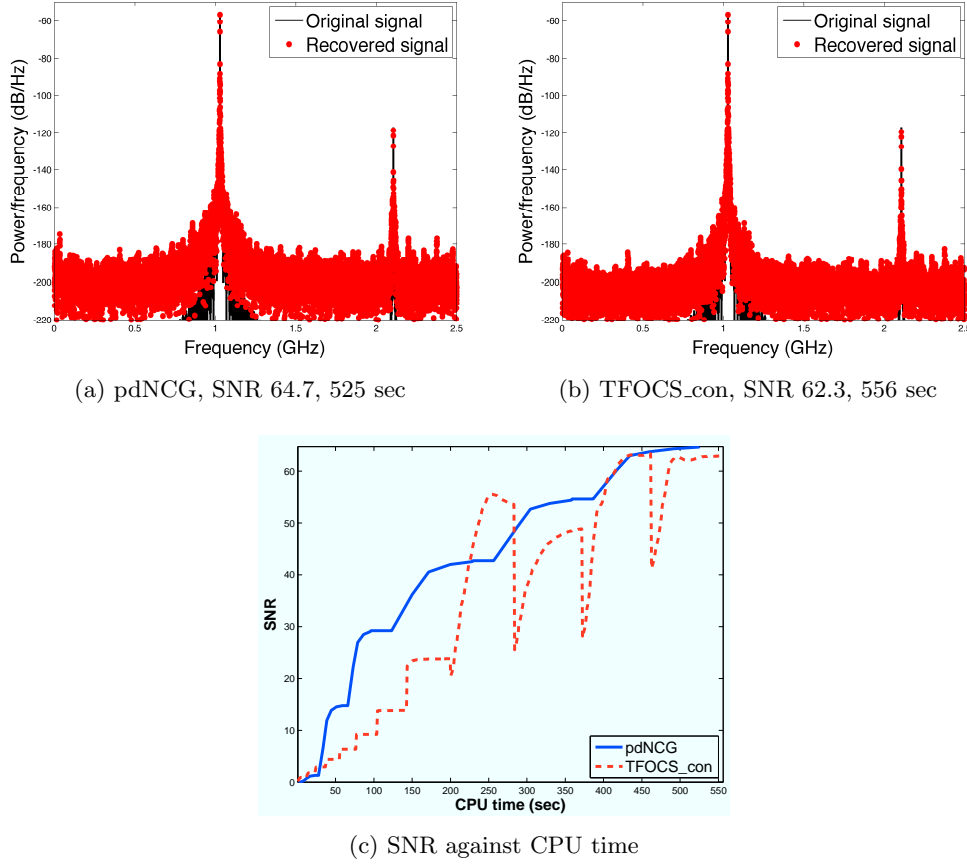


Fig. 7.4: Reconstruction of two radio-frequency radar tones by TFOCS_con and pdNCG. In Figure 7.4a the reconstructed signal by pdNCG is shown. The signal has SNR 64.7 dB and it required 525 seconds of CPU time to be reconstructed. In Figure 7.4b the reconstructed signal by TFOCS_con is shown. The signal has PSNR 62.3 dB and it required 556 seconds of CPU time to be reconstructed. In Figure 7.4c we present the SNR for each iteration of pdNCG and TFOCS_con against CPU time.

- [10] R. H. Chan, T. F. Chan, and H. M. Zhou. Advanced signal processing algorithms. in *Proceedings of the International Society of Photo-Optical Instrumentation Engineers*, F. T. Luk, ed., SPIE, pages 314–325, 1995.
- [11] T. F. Chan, G. H. Golub, and P. Mulet. A nonlinear primal-dual method for total variation-based image restoration. *SIAM J. Sci. Comput.*, 20(6):1964–1977, 1999.
- [12] L. Condat. A generic proximal algorithm for convex optimization - Application to total-variation. *IEEE Signal Processing Letters*, 21(8):985–989, 2014.
- [13] M. F. Duarte, M. A. Davenport, D. Takhar, J. N. Laska, S. Ting, K. F. Kelly, and R. G. Baraniuk. Single-pixel imaging via compressive sampling. *IEEE Signal Processing Magazine*, 25(2):83–91, 2008. <http://dsp.rice.edu/cscamera>.
- [14] J. E. Esser. *Primal Dual Algorithms for Convex Models and Applications to Image Restoration, Registration and Nonlocal Inpainting*. PhD thesis, University of California, 2010.
- [15] K. Fountoulakis and J. Gondzio. A second-order method for strongly convex ℓ_1 -regularization problems. *Mathematical Programming (accepted)*, 2015. DOI: 10.1007/s10107-015-0875-4.

- [16] K. Fountoulakis, J. Gondzio, and P. Zhlobich. Matrix-free interior point method for compressed sensing problems. *Mathematical Programming Computation*, 6(1):1–31, 2014.
- [17] E. T. Hale, W. Yin, and Y. Zhang. Fixed-point continuation for ℓ_1 -minimization: Methodology and convergence. *SIAM J. Optim.*, 19(3):1107–1130, 2008.
- [18] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.
- [19] C. Li, W. Yin, H. Jiang, and Y. Zhang. An efficient augmented Lagrangian method with applications to total variation minimization. *Computational Optimization and Applications*, 56(3):507–530, 2013.
- [20] I. Loris and C. Verhoeven. On a generalization of the iterative soft-thresholding algorithm for the case of non-separable penalty. *Inverse Problems*, 27(12):1–15, 2011.
- [21] S. Mallat. A wavelet tour of signal processing, second ed. *Academic Press, London*, 1999.
- [22] J. J. Moreau. Proximité et dualité dans un espace Hilbertien. *Bull. Soc. Math. France*, 93:273–299, 1965.
- [23] D. Needell and R. Ward. Stable image reconstruction using total variation minimization. *SIAM J. Imaging Sciences*, 6(2):1035–1058, 2013.
- [24] Y. Nesterov. Smooth minimization of non-smooth functions. *Math. Program.*, 103(1):127–152, 2004.
- [25] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, New York, 2006.
- [26] S. Vaiteer, G. Peyré, C. Dossal, and J. Fadili. Robust sparse analysis regularization. *IEEE Trans. Inf. Theory*, 59(4):2001–2016, 2013.
- [27] C. R. Vogel and M. E. Oman. Fast, robust total variation-based reconstruction of noisy, blurred images. *Image Processing, IEEE Transactions on*, 7(6):813–824, 1998.

Appendix A. Continuous path. In the following lemma we show that $x_{c,\mu} := \arg \min f_c^\mu(x)$ (f_c^μ is defined in (2.2)) for c constant is a continuous and differentiable function of μ .

LEMMA A.1. *Let c be constant and consider $x_{c,\mu}$ as a functional of μ . If condition (1.4) is satisfied, then $x_{c,\mu}$ is continuous and differentiable.*

Proof. The optimality conditions of problem (2.2) are

$$c\nabla\psi_\mu(W^*x) + A^\top(Ax - b) = 0.$$

According to definition of $x_{c,\mu}$, we have

$$\begin{aligned} c\nabla\psi_\mu(W^*x_{c,\mu}) + A^\top(Ax_{c,\mu} - b) &= 0 & \implies \\ c\frac{d\nabla\psi_\mu(W^*x_{c,\mu})}{d\mu} + A^\top A\frac{dx_{c,\mu}}{d\mu} &= 0 & \implies \\ c\left(\nabla^2\psi_\mu(W^*x_{c,\mu})\frac{dx_{c,\mu}}{d\mu} + \frac{d\nabla\psi_\mu(W^*x)}{d\mu}\Big|_{x_{c,\mu}}\right) + A^\top A\frac{dx_{c,\mu}}{d\mu} &= 0 & \iff \\ \left(c\nabla^2\psi_\mu(W^*x_{c,\mu}) + A^\top A\right)\frac{dx_{c,\mu}}{d\mu} + c\frac{d\nabla\psi_\mu(W^*x)}{d\mu}\Big|_{x_{c,\mu}} &= 0 & \iff \\ \nabla^2 f_c^\mu(W^*x_{c,\mu})\frac{dx_{c,\mu}}{d\mu} + c\frac{d\nabla\psi_\mu(W^*x)}{d\mu}\Big|_{x_{c,\mu}} &= 0, \end{aligned}$$

where $d\nabla\psi_\mu(W^*x)/d\mu|_{x_{c,\mu}}$ is the first-order derivative of $\nabla\psi_\mu(W^*x)$ as a functional of μ , measured at $x_{c,\mu}$. Notice that due to condition $\text{Ker}(W^*) \cap \text{Ker}(A) = \{0\}$ we have that $\nabla^2 f_c^\mu(x)$ is positive definite $\forall x$, hence $x_{c,\mu}$ is unique. Therefore, the previous system has a unique solution, which means that $x_{c,\mu}$ is uniquely differentiable as a functional of μ with c being constant. Therefore, $x_{c,\mu}$ is continuous as a functional of μ . \square

REMARK A.2. *Lemma A.1 and continuity imply that there exists sufficiently small smoothing parameter μ such that $\|x_{c,\mu} - x_c\|_2 < \omega$ for any arbitrarily small $\omega > 0$.*